



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

**TUGAS AKHIR - TE 141599**

**PERMODELAN KURVA KARAKTERISTIK INVERSE NON-  
STANDART PADA RELE ARUS LEBIH DENGAN METODE  
ADAPTIVE NEURO FUZZY INFERENCE SYSTEM (ANFIS)**

Erhankana Ardiana P.  
NRP 2214105085

Dosen Pembimbing  
Dr. Ir. Margo Pujiantara, M.T.  
Dr. Eng. Ardyono Priyadi, S.T., M. Eng.

JURUSAN TEKNIK ELEKTRO  
Fakultas Teknologi Industri  
Institut Teknologi Sepuluh Nopember  
Surabaya 2016



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

**FINAL PROJECT - TE 141599**

***Non-standard Inverse Characteristic Curve Modeling of  
Overcurrent Relay with Adaptive Neuro Fuzzy Inference  
System***

Erhankana Ardiana P.  
NRP 2214105085

Advisor  
Dr. Ir. Margo Pujiantara, M.T.  
Dr. Eng. Ardyono Priyadi, S.T., M. Eng.

ELECTRICAL ENGINEERING DEPARTMENT  
Faculty Of Industrial Technology  
Sepuluh Nopember Institute Of Technology  
Surabaya 2016



**PERMODELAN KURVA KARAKTERISTIK INVERSE NON-  
STANDART PADA RELE ARUS LEBIH DENGAN METODE  
ADAPTIVE NEURO FUZZY INFERENCE SYSTEM (ANFIS)**

**TUGAS AKHIR**

**Diajukan Guna Memenuhi Sebagian Persyaratan  
Untuk Memperoleh Gelar Sarjana Teknik**

**Pada**

**Bidang Studi Teknik Sistem Tenaga  
Jurusan Teknik Elektro**

**Institut Teknologi Sepuluh Nopember**

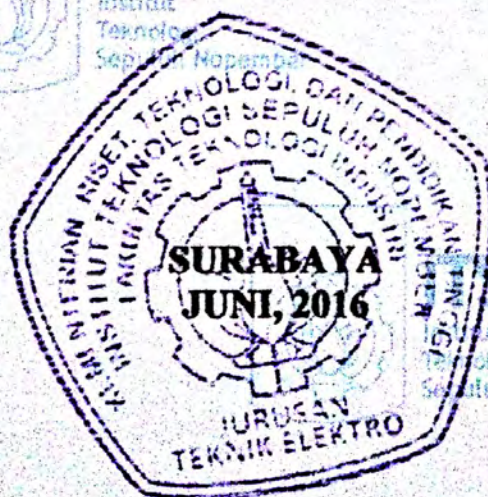
**Menyetujui:**

**Dosen Pembimbing I**

**Dosen Pembimbing II**

**Dr. Ir. Margo Pujiانتara, MT.**  
**NIP. 196603181990101001**

**Dr.Eng.Ardyono Priyadi, ST.,M.Eng**  
**NIP. 197309271998031004**





***“halaman ini sengaja dikosongkan”***

**LEMBAR PERNYATAAN  
PERSETUJUAN PUBLIKASI KARYA ILMIAH  
UNTUK KEPENTINGAN AKADEMIS**

Sebagai mahasiswa Institut Teknologi Sepuluh Nopember Surabaya, yang bertanda tangan di bawah ini saya :

Nama : Erhankana Ardiana Putra  
Nrp. : 2214105085  
Jurusan / Fak. : Teknik Elektro / FTI  
Alamat kontak : Kalijudan VII No.8 Surabaya  
a. Email : erhankana@gmail.com  
b. Telp/HP : 0821 8880 7707

Menyatakan bahwa semua data yang saya *upload* di Digital Library ITS merupakan hasil final (revisi terakhir) dari karya ilmiah saya yang sudah disahkan oleh dosen penguji. Apabila dikemudian hari ditemukan ada ketidaksesuaian dengan kenyataan, maka saya bersedia menerima sanksi.

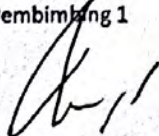
Demi perkembangan ilmu pengetahuan, saya menyetujui untuk memberikan **Hak Bebas Royalti Non-Eksklusif (Non-Exclusive Royalti-Free Right)** kepada Institut Teknologi Sepuluh Nopember Surabaya atas karya ilmiah saya yang berjudul :

Permodelan Kurva Karakteristik Inverse non-standart pada rele arus lebih dengan metode ANFIS

Dengan Hak Bebas Royalti Non-Eksklusif ini, Institut Teknologi Sepuluh Nopember Surabaya berhak menyimpan, mengalih-media/format-kan, mengelolanya dalam bentuk pangkalan data (*database*), mendistribusikannya, dan menampilkan/mempublikasikannya di internet atau media lain untuk kepentingan akademis tanpa meminta ijin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta. Saya bersedia menanggung secara pribadi, segala bentuk tuntutan hukum yang timbul atas pelanggaran Hak Cipta dalam karya Ilmiah saya ini tanpa melibatkan pihak Institut Teknologi Sepuluh Nopember Surabaya.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dosen Pembimbing 1

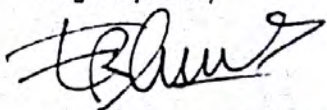
  
Dr. Ir. Marga Pujiantara, M.T.

NIP. 1966 0318 1990 10 1001

Dibuat di : Surabaya

Pada tanggal : 29 Juni 2016

Yang menyatakan,

  
Erhankana A.P.

Nrp. 2214105085

**KETERANGAN :**

Tanda tangan pembimbing wajib dibubuhi stempel jurusan.

Form dicetak dan diserahkan di bagian Pengadaan saat mengumpulkan hard copy TA/Tesis/Disertasi.

# PERMODELAN KURVA KARAKTERISTIK INVERSE NON-STANDART PADA RELE ARUS LEBIH DENGAN METODE ADAPTIVE NEURO FUZZY INFERENCE SYSTEM (ANFIS)

Erhankana Ardiana Putra  
NRP 2214105085

## Dosen Pembimbing

Dr. Ir. Margo Pujiantara, MT.  
Dr. Ardyono Priyadi, ST.,M.Eng

## ABSTRAK

Pada sistem kelistrikan terutama pada sistem proteksi kelistrikan dewasa ini sangat dibutuhkan sistem yang handal, sehingga perkembangan pada sistem proteksi sudah semakin maju dengan adanya penggunaan rele digital. Rele digital digunakan dengan mempertimbangkan kecepatan, keakuratan dan serta flexible dalam sistem koordinasi. Flexibilitas ini dimaksudkan bahwa rele digital dapat digunakan menjadi rele arus lebih (*overcurrent relay*) sesuai pembahasan tugas akhir ini dan dapat disetting menurut keinginan *user* sesuai karakteristik kurva OCR konvensional/standart (normal inverse, very inverse, long time inverse, extreme inverse) yang akan digunakan dalam koordinasi. Jenis kurva pada rele digital juga dapat disetting diluar rumus kurva konvensional/standart yang seperti sudah disebutkan sebelumnya, kurva diluar rumusan standart disebut kurva rele *non-standart*. Kurva rele non-standart digunakan untuk memudahkan pengguna untuk menentukan waktu trip berdasarkan arus yang diinginkan dan sebagai solusi jika pada koordinasi proteksi mengalami kendala dalam koordinasi kurva rele. Pada tugas akhir ini akan dibahas bagaimana membuat atau memodelkan kurva karakteristik *inverse overcurrent* rele *non-standart* dengan menggunakan metode (Adaptive Neuro Fuzzy Inference System) atau biasa disebut metode pembelajaran ANFIS. Kurva *non-standart* didapatkan dengan pengambilan titik-titik data baru berupa arus dan waktu *trip* sesuai keinginan *user*. Data baru tersebut akan digabungkan dengan data lama sehingga menghasilkan data *non-standart* yang nantinya akan dilakukan pembelajaran dengan metode ANFIS untuk mendapatkan desain kurva *non-standart*. Setelah didapatkan desain kurva non-standart akan dilakukan pengujian keakuratan dengan

mengganti nilai MF (membership function) didapatkan hasil rata-rata error terkecil 2,56% (MF=10 dan epoch=100). Pengujian selanjutnya dengan mengubah nilai epoch didapatkan nilai keakuratan dengan error terkecil pada epoch = 500 (MF=4). Simulasi pada software Etap 12.6 untuk memastikan bahwa kurva *non-standart* sudah sesuai dengan koordinasi proteksi yang diinginkan dan dapat diaplikaikan pada software Etap 12.6 dengan nilai error rata-rata 0,7%.

Key word : *ANFIS, rele digital, overcurrent rele, non-standart, inverse overcurrent rele*

# ***NON-STANDARD INVERSE CHARACTERISTIC CURVE MODELING OF OVERCURRENT RELAY WITH ADAPTIVE NEURO FUZZY INFERENCE SYSTEM***

Erhankana Ardiana Putra  
NRP 2214105085

## **Counsellor Lecturer**

Dr. Ir. Margo Pujiantara, MT.  
Dr. Ardyono Priyadi, ST.,M.Eng

## ***ABSTRACT***

In electrical especially at protection system electrical nowadays very needed reliable system, so that development for protection system getting ahead with their use of digital relays. Digital relay used consider about fast respon, accuracy and flexibility at coordination system protection. Flexibility means that digital relay can be use to be overcurrent relay due to this study of final project and can be set obey by user consider by curve characteristic conventional/standart OCR (normal inverse, very inverse, long time inverse, extreme inverse) that will be use to coordination. Type of curve digital relay can be set outside of the curve conventional/standart formula that already mentioned before, curve that outside from standart formula called non-standart curve relay. Non-standart curve relay be used for easier user for decide time trip according by current and non-standart curve can be an solution if coordination protection have problem at coordination curve. This final project will discuss about how to make or modeling characteristic curve inverse overcurrent relay non-standart use ANFIS method. Non-standart curve can be get by taking new node that form like current and time trip data. That new data will be combine with old data so that produce a new data and can be called non-standart data. ANFIS will learning Non-standart data and try to get deign non-standart curve. After getting non-standart curve that will do some accurate test like change number of membership function (MF) and get result smallest mean error 2,56% (MF=10 and epoch = 100). Next testing is change number of epoch (iteration) and the result smallest error at epoch = 500 (MF=4). Simulation at Etap 12.6 for make sure if that non-standart curve can be suitable with coordination protection or can be aplicated at Etap 12.6 and the result mean error between result at Etap and learning ANFIS 0,7% .



Key word : *ANFIS, digital relay, overcurrent relay, non-standart, inverse overcurrent relay*

## DAFTAR ISI

JUDUL .....	i
LEMBAR PENGESAHAN .....	iii
ABSTRAK .....	vii
<i>ABSTRACT</i> .....	ix
KATA PENGANTAR .....	xi
DAFTAR ISI .....	xiii
DAFTAR GAMBAR .....	xv
DAFTAR TABEL.....	xvii

## BAB I. PENDAHULUAN

1.1 Latar Belakang.....	1
1.2 Perumusan Masalah .....	2
1.3 Batasan Masalah .....	2
1.4 Tujuan dan Masalah .....	2
1.5 Metodologi .....	2
1.6 Sistematika Penulisan .....	5
1.7 Relevansi .....	5

## BAB II. TINJAUAN PUSTAKA

2.1 Gangguan pada Sistem Tenaga Listrik .....	7
2.2 Jenis-jenis Gangguan .....	7
2.2.1 Gangguan Beban Lebih ( <i>overload</i> ) .....	8
2.2.2 Gangguan Hubung Singkat.....	8
2.2.3 Kontribusi Arus Hubung Singkat .....	9
2.2.4 Perhitungan Arus Hubung Singkat .....	9
2.3 Sifat-sifat Gangguan .....	11
2.4 Trafo Arus (Current Transformer) .....	11
2.5 Rele pengaman pada sistem tenaga listrik .....	11
2.6 Rele Arus Lebih .....	13
2.6.1 Rele arus lebih waktu inverse.....	13
2.7 Rele karakteristik Tripping Non-Standard.....	14
2.8 Adaptive neuro fuzzy inference system (ANFIS) .....	15
2.8.1 Gambaran umum ANFIS .....	15
2.8.2 Arsitektur ANFIS.....	16
2.8.3 Algoritma pembelajaran hybrid .....	19

<b>BAB III. PERANCANGAN SOFTWARE DAN HARDWARE</b>	
3.1 Perancangan software .....	21
3.1.1 Pengambilan data rele pada Etap 12.6 .....	21
3.1.2 Perancangan interface GUI Matlab 2013 .....	29
3.2 Perancangan hardware dan Integrasi .....	31
3.2.1 Perancangan Hardware .....	31
<b>BAB IV. HASIL SIMULASI DAN ANALISIS</b>	
4.1 Pengambilan data rele.....	34
4.2 Pembuatan Interface GUI Matlab.....	37
4.3 Pengujian data kurva normal inverse.....	38
4.3.1 Pengujian keakuratan dengan mengganti jumlah MF	42
4.4 Pengambilan data non-standart .....	47
4.5 Pengujian data kurva non-standart .....	50
4.5.1 Pembuatan data non-standart .....	50
4.5.2 Pengujian learning ANFIS data non-standart.....	53
4.5.3 Pengujian keakuratan dengan mengganti jumlah MF	56
4.5.4 Pengujian keakuratan dengan mengganti jumlah Epoch .....	62
4.6 Pengaplikasian kurva pada software Etap 12.6 .....	66
4.6.1 Perbandingan hasil dari data non-standart dan hasil Etap12.6.....	71
4.7 Pengujian pada Hardware .....	73
<b>BAB V. PENUTUP</b>	
5.1 Kesimpulan .....	79
5.2 Saran .....	79
<b>DAFTAR PUSTAKA .....</b>	81
<b>LAMPIRAN .....</b>	83
<b>BIODATA PENULIS.....</b>	119



## DAFTAR TABEL

Tabel 3.1 Data rele PT.HESS Indonesia .....	21
Tabel 3.2 Kondisi normal setting rele R-VCB 9.....	25
Tabel 3.3 Kondisi normal setting rele R-ACB 2.....	25
Tabel 3.4 Setting rele R-VCB-9 saat kondisi berhimpitan.....	26
Tabel 4.1 Data kurva normal inverse (standart IEC) .....	36
Tabel 4.2 Data dengan nilai jarak antara arus sebesar 50 A.....	39
Tabel 4.3 Hasil error pegujian data test normal inverse .....	42
Tabel 4.4 Hasil error perbandingan trip ANFIS dan trip target dengan MF = 3 .....	43
Tabel 4.5 Hasil error perbandingan trip ANFIS dan trip target dengan MF = 5.....	45
Tabel 4.6 Hasil error perbandingan trip ANFIS dan trip target dengan MF = 10.....	46
Tabel 4.7 Tabel data baru dalam bentuk excel.....	50
Tabel 4.8 Data Kurva Normal Inverse .....	51
Tabel 4.9 Data Baru sebanyak 10 data.....	51
Tabel 4.10 Data non-standart .....	52
Tabel 4.11 Hasil error trip ANFIS berbanding trip data non-standart	55
Tabel 4.12 Hasil error perbandingan trip ANFIS dan trip target dengan MF = 3.....	58
Tabel 4.13 Hasil error perbandingan trip ANFIS dan trip target dengan MF = 5.....	59
Tabel 4.14 Hasil error perbandingan trip ANFIS dan trip target dengan MF = 10.....	61
Tabel 4.15 Data percobaan non-standart.....	62
Tabel 4.16 Data non-standart .....	67
Tabel 4.17 Error perbandingan hasil Etap dan data non-standart.....	72
Tabel 4.18 Data non-standart untuk pengujian hardware.....	73
Tabel 4.19 Data non-standart dalam bentuk IL / Is .....	74
Tabel 4.20 Error perbandingan trip ANFIS dan trip Hardware.....	77



***“halaman ini sengaja dikosongkan”***



## DAFTAR GAMBAR

Gambar 1.1 Metodologi Tugas Akhir .....	4
Gambar 2.1 Alur kerja rele saat ada gangguan .....	12
Gambar 2.2 Karakteristik kurva inverse pada standart IEC .....	14
Gambar 2.3 Kurva rele standart dan kurva rele yang diinginkan .....	15
Gambar 2.4 Aturan arsitektur ANFIS .....	16
Gambar 2.5 ANFIS dengan model sugeno .....	17
Gambar 2.6 Arsitektur jaringan ANFIS .....	17
Gambar 3.1 Single line diagram PT.HESS Indonesia .....	23
Gambar 3.2 Kurva rele R-VCB 9 saat kondisi normal.....	24
Gambar 3.3 Setting ulang rele R-VCB 9.....	26
Gambar 3.4 Kurva rele R-VCB 9 kondisi berhimpitan .....	27
Gambar 3.5 Kurva rele dengan titik data baru .....	28
Gambar 3.6 Interface GUI Matlab 2013 .....	29
Gambar 3.7 Flowchart permodelan kurva non-standart .....	30
Gambar 3.8 Mikrokontroller ARM .....	31
Gambar 3.9 LCD 128x64 .....	32
Gambar 4.1 Single line diagram dan koordinasi rele .....	34
Gambar 4.2 Setting rele R-VCB 9 di Etap 12.6 .....	35
Gambar 4.3 Workspace GUI Matlab 2013 .....	37
Gambar 4.4 Tampilan interface GUI .....	38
Gambar 4.5 Interface GUI percobaan data jarak antar arus 50 A ....	40
Gambar 4.6 Plot kurva hasil training.....	40
Gambar 4.7 Pengujian data test sesuai data training .....	41
Gambar 4.8 Pengujian data test diluar data training .....	41
Gambar 4.9 Hasil percobaan MF 3 pada interface GUI.....	42
Gambar 4.10 Hasil plot kurva ANFIS MF=3.....	43
Gambar 4.11 Hasil plot kurva learning ANFIS MF=5 .....	44
Gambar 4.12 Hasil plot kurva learning ANFIS MF = 10 .....	46
Gambar 4.13 Tampilan GUI dengan plot kurva .....	48
Gambar 4.14 Tombol "get data" pada interface GUI.....	48
Gambar 4.15 Pengambilan titik-titik data baru .....	49
Gambar 4.16 Tombol save data baru di interface GUI.....	49
Gambar 4.17 Pengujian data non-standart pada interface GUI .....	53
Gambar 4.18 Tampilan kurva non-standart pada kolom plot interface GUI .....	54
Gambar 4.19 Pengujian hasil trip dengan arus 130,9157 A .....	55
Gambar 4.20 Hasil percobaan MF 3 pada interface GUI .....	57
Gambar 4.21 Hasil plot kurva <i>learning</i> ANFIS MF = 3 .....	57
Gambar 4.22 Hasil plot kurva <i>learning</i> ANFIS MF = 5 .....	59

Gambar 4.23 Hasil plot kurva <i>learning</i> ANFIS MF = 10 .....	60
Gambar 4.24 Hasil plot kurva <i>learning</i> ANFIS Epoch = 100 .....	63
Gambar 4.25 Hasil data test arus = 693,3 A, Epoch 100 .....	63
Gambar 4.26 Pembuktian plot dengan data test arus = 693,3 A Epoch 100 .....	64
Gambar 4.27 Hasil data test arus = 693,3 A, Epoch 250 .....	64
Gambar 4.28 Pembuktian plot dengan data test arus = 693,3 A Epoch 250 .....	65
Gambar 4.29 Hasil data test arus = 693,3 A, Epoch 500 .....	65
Gambar 4.30 Pembuktian plot dengan data test arus = 693,3 A Epoch 500 .....	66
Gambar 4.31 Propertise "USER CURVE" dan data tabel memasukkan data .....	68
Gambar 4.32 Hasil plot data non-standart pada Etap 12.6.....	69
Gambar 4.33 Plot kurva saat kondisi beringgungan .....	70
Gambar 4.34 Hasil waktu trip data non-standart pada Etap 12.6.....	71
Gambar 4.35 Interface pada visual basic .....	74
Gambar 4.36 Parameter premis dan konsekuen.....	75
Gambar 4.37 Alur percobaan pada hardware .....	76

# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

Kehandalan suatu sistem kelistrikan merupakan suatu hal yang menjadi prioritas utama. Terutama pada sistem distribusi yang menjadi pemasok aliran daya menuju beban. Pentingnya menjaga kehandalan sistem distribusi guna mempertahankan kontinuitas tenaga listrik menuju beban.

Salah satu cara dalam mempertahankan kehandalan ialah koordinasi sistem proteksi yang baik pada sistem kelistrikan tersebut. Banyaknya jenis rele yang digunakan pada sistem tentunya harus di-koordinasikan secara tepat agar mampu menanggulangi setiap gangguan yang terjadi tanpa kesalahan sedikit-pun. Pada metode koordinasi rele sebelumnya telah memberikan hasil yang baik, namun tentunya diperlukan metode lain yang mampu memberikan hasil yang optimal. Salah satunya metode karakteristik kurva *tripping* non-standar, dimana kurva rele non-standart ini dapat bermanfaat saat terjadi kurva yang berhimpitan atau saling *cross* sehingga menimbulkan trip pada CB (*Circuit Breaker*) yang bersamaan pada beberapa arus tertentu. Sehingga dibutuhkan permodelan kurva non-standart dengan maksud membuat kurva baru dengan mengambil data baru pada plot kurva sehingga terbentuk kurva baru yang sudah tidak berhimpitan satu sama lain, kurva non-standart juga mampu mempercepat waktu *tripping* maksimum dan rata rata yang efektif sesuai dengan kurva karakteristik masing-masing rele.

Dewasa ini sistem digital dapat diaplikasikan sebagai rele proteksi yang dimana rele berbasiskan sistem digital mempunyai beberapa keunggulan. Keunggulan rele proteksi digital ini adalah respon yang cepat terhadap gangguan, keakuratan dalam perhitungan, fleksibel serta dapat berkoordinasi dengan baik<sup>[1]</sup>. Digital rele proteksi ini dapat dirancang dengan sistem berbasis mikrokontroller dengan dikombinasikan dengan kemampuan belajar ANFIS. Dengan menggunakan kemampuan belajar ANFIS dapat dihasilkan sebuah rele proteksi digital dengan tingkat keakuratan yang tinggi. Dalam tugas akhir ini akan dilakukan studi mengenai cara memodelkan kurva non-standart menggunakan metode ANFIS. Mendesain kurva non-standart pada tugas akhir ini menerapkan kemampuan belajar *Adaptive Neuro Fuzzy Interference System* (ANFIS). Laptop atau *Personal Computer* (PC) digunakan sebagai sarana untuk membuat interface pemodelan kurva non-standart serta untuk simulasi hasil dari permodelan kurva menggunakan metode ANFIS yang dimana didalam proses permodelan



metode ANFIS dilakukan inisialisasi beserta learning hingga didapat hasil learning dan parameter hasil learning.

## 1.2 Perumusan Masalah

Perumusan masalah yang akan dibahas pada tugas akhir ini adalah :

1. Bagaimana cara mendapatkan data baru untuk membentuk kurva non-standart.
2. Memodelkan kurva non-standart melalui laptop atau PC.
3. Melakukan learning pada data kurva non-standart.
4. Menguji keakuratan learning ANFIS dengan beberapa arus yang diinginkan.
5. Pengaplikasian kurva non-standart pada software Etap 12.6.
6. Pengaplikasian kurva non-standart pada hardware *Digital Relay Protection* (DPR).

## 1.3 Batasan Masalah

Dalam pengerjaan tugas akhir ini, permasalahan dibatasi sebagai berikut :

1. Gangguan yang terjadi hanya gangguan hubung singkat.
2. Jenis peralatan pengaman yang digunakan adalah rele arus lebih (Over Current Relay).
3. Kurva rele yang akan dimodelkan berfokus pada bentuk *inversenya*.
4. *Software* yang digunakan untuk membuat interface adalah MATLAB 2013.
5. Pengujian dilakukan pada interface GUI MATLAB 2013
6. Pengujian hasil hanya melihat dari waktu trip permodelan kurva dengan beberapa arus yang diinginkan user.

## 1.4 Tujuan Tugas Akhir

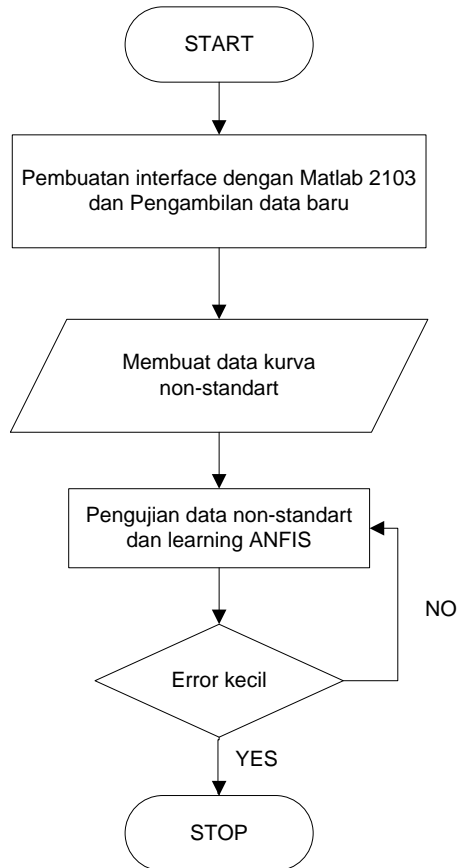
Tugas akhir ini bertujuan untuk memodelkan kurva non-standart dengan metode ANFIS dan mendapatkan nilai *tripping* yang akurat pada rele, sehingga mampu meningkatkan selektifitas rele saat terjadi gangguan dan mengatasi masalah saat adanya plot rele yang saling berhimpitan. Serta mempercepat waktu *tripping* dan meningkatkan keandalan sebuah pengaman peralatan.

## 1.5 Metodologi

Metodologi penelitian yang dilakukan dalam tugas akhir ini adalah sebagai berikut :

1. Studi literature  
Studi literature dilakukan untuk memperoleh teori pendukung yang berasal dari *paper*, jurnal dan *text book*. Dalam literature tersebut dipelajari teori – teori tentang rele pengaman digital dan devinisi kurva non-standart. Selain itu akan mempelajari metode ANFIS dan cara membuat *interface* permodelan kurva non-standart.
2. Pemodelan dan Perancangan sistem  
Penulis melakukan pembuatan program desain kurva dengan menggunakan *software* MATLAB 2013 dan perancangan *interface* permodelan kurva menggunakan GUI MATLAB 2013. Pada bagian ini penulis juga akan melakukan percobaan pengambilan beberapa data baru yang berupa waktu *trip* rele melalui *interface* GUI Matlab 2103. Dari data baru tersebut akan digabungkan dengan data kurva conventional/standart yang sudah ada dimana beberapa data akan digantikan dengan data baru yang didapatkan, dan dilakukan pengujian.
3. Pengujian kurva non-standart  
Penulis akan menguji hasil dari waktu *trip* data kurva yang baru atau non-standart dan membandingkannya dengan hasil *trip* rele *learning* ANFIS. Dari hasil trip akan didapatkan error antara *trip* kurva non-standart dan *learning* ANFIS.
4. Analisa Data  
Pada tahap ini dilakukan analisa dari hasil pengujian hasil *trip* dan akan didapatkan error antara *trip* kurva non-standart dan learning ANFIS. Apakah hasil dari *trip* kurva non-standart dan learning ANFIS sesuai dengan acuan nilai rata-rata error yang kecil antara hasil *trip* kurva dan *learning* ANFIS. Dalam tahap ini juga dialkukan uji keakuratan dengan mengganti nilai MF (*memberhip function*) dan nilai *epoch* (iterasi).
5. Kesimpulan  
Dari hasil pengujian dengan jumlah data yang berbeda-beda dilakukan analisa dan penarikan kesimpulan, saran atau rekomendasi tentang penelitian yang telah dilakukan.

Skema flowchart pengerjaan tugas akhir, sebagai berikut :



**Gambar 1.1** Metodologi Tugas Akhir

## **1.6 Sistematika Penulisan**

Laporan Tugas Akhir ini disusun dalam suatu sistematika sebagai berikut :

### **BAB I : PENDAHULUAN**

Bab ini terdiri dari latar belakang, perumusan masalah, tujuan, metodologi pengerjaan tugas akhir, sistematika pembahasan dan relevansi dari penulis.

### **BAB II : PROTEKSI DAN METODE ANFIS**

Pada bab ini menjelaskan teori-teori penunjang yang digunakan dalam pengerjaan tugas akhir ini.

### **BAB III : PERANCANGAN SOFTWARE.**

Dalam bab ini menjelaskan perancangan konfigurasi sistem. Perancangan software yang dibuat mengimplemantasikan rumusan baru kurva non-standart pada metoda ANFIS.

### **BAB IV : HASIL PENGUJIAN DAN ANALISIS**

Dalam bab ini dibahas mengenai hasil pengujian data kurva non-standart. Pengujian ini dilakukan dengan merubah variable yang berkaitan dengan keakuratan data. Selain menguji data pada bagian ini dilakukan pengujian kurva rele non-standart pada software Etap 12.6. Saat pengujian dilakukan perhitungan rata-rata error hasil *trip* kurva non-standart dan *learning* ANFIS. Dilakukan juga analisa pengujian pada hardware.

### **BAB V : PENUTUP**

Bab ini berisi kesimpulan dan saran dari hasil pengujian data, studi literatur dan analisis yang telah dilakukan.

## **1.7 Relevansi**

Dari hasil pengerjaan tugas akhir ini diharapkan dapat memberikan kontribusi terhadap perkembangan sistem kelistrikan terutama pada sistem proteksi.

***“halaman ini sengaja dikosongkan”***

## **BAB II**

### **HUBUNG SINGKAT, RELE ARUS LEBIH & ANFIS**

#### **2.1 Gangguan pada Sistem Tenaga Listrik**

Gangguan yang terjadi pada sistem tenaga listrik sangat beragam besaran dan jenisnya. Gangguan dalam sistem tenaga listrik adalah keadaan tidak normal dimana keadaan ini dapat mengakibatkan terganggunya kontinuitas pelayanan tenaga listrik. Secara umum klasifikasi gangguan pada system tenaga listrik disebabkan 2 faktor, yaitu <sup>[1]</sup> :

1. Gangguan berasal dari system
  2. Gangguan berasal dari luar system
- 
1. Penyebab gangguan yang berasal dari dalam system antara lain:
    - Tegangan dan arus abnormal.
    - Pemasangan yang kurang baik.
    - Kesalahan mekanis karena proses penuaan.
    - Beban lebih.
    - Kerusakan material seperti isolator pecah, kawat putus, atau kabel cacat isolasinya.
  2. Sedangkan untuk gangguan yang berasal dari luar system antara lain:
    - Gangguan – gangguan mekanis karena pekerjaan galian saluran lain. Gangguan ini terjadi untuk system kelistrikan bawah tanah.
    - Pengaruh cuaca seperti hujan, angin, surja petir. Pada gangguan surja petir. Pada gangguan surja petir dapat mengakibatkan gangguan tegangan lebih dan dapat menyebabkan gangguan hubung singkat karena tembus isolasi peralatan(breakdown).
    - Pengaruh lingkungan seperti pohon, binatang dan benda-benda asing serta akibat kecerobohan manusia

#### **2.2 Jenis – Jenis Gangguan**

Adanya gangguan pada sistem tenaga listrik menyebabkan terjadinya kenaikan arus yang sangat besar menuju ke titik gangguan, sehingga arus yang melewati peralatan juga akan sangat besar. Bila hal

ini dibiarkan maka arus yang melewati peralatan dapat merusak peralatan karena diluar kapasitas yang diijinkan. Pada umumnya gangguan yang terjadi pada sistem adalah gangguan beban lebih (*overload*) dan gangguan hubung singkat (*Short Circuit*).

### 2.2.1 Gangguan Beban Lebih (*Overload*)

Gangguan ini merupakan gangguan yang timbul akibat adanya beban yang berlebih sehingga mengakibatkan arus yang menuju beban melebihi arus nominal ( $I > I_n$ ). Arus yang mengalir melebihi kapasitas ini jika tidak segera diatasi, maka dapat merusak peralatan listrik tersebut.

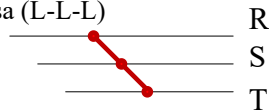
### 2.2.2 Gangguan Hubung Singkat

Hubung singkat adalah terjadinya hubungan penghantar bertegangan atau penghantar tidak bertegangan yang memiliki beda potensial secara langsung, sehingga terjadi aliran arus yang tidak normal atau sangat besar. Gangguan hubung singkat ini dapat dibedakan menjadi 2 kategori, yaitu:

#### 1. Gangguan hubung singkat simetri

##### a. Hubung singkat 3 fasa (L-L-L)

Gambar rangkaian :



##### b. Hubung singkat 3 fasa ke tanah (L-L-L-G)

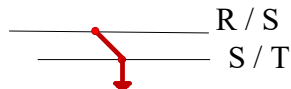
Gambar rangkaian :



#### 2. Gangguan hubung singkat asimetri

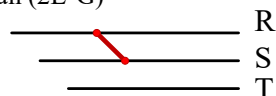
##### a. Hubung singkat 2 fasa (L-L)

Gambar rangkaian :



##### b. Hubung singkat 2 fasa ke tanah (2L-G)

Gambar rangkaian :





- c. Hubung singkat 1 fasa ke tanah (1L-G)

Gambar rangkaian :



### 2.2.3 Kontribusi Arus Hubung Singkat

Nilai arus hubung singkat berasal dari besar sumber pembangkitkan sistem, yang mana merupakan nilai dari reaktansi sumber, peralatan, serta sistem keseluruhan hingga ke titik gangguan. Sumber arus hubung singkat berasal dari sistem *utility*, generator, motor sinkron dan motor induksi.

Terdapat 3 kondisi peralihan ketika terjadi hubung singkat (*short circuit*):

1. Kondisi pertama dimana *short circuit* terjadi pada kondisi  $\frac{1}{2}$  cycle. Kondisi ini disebut periode *subtransient*, yang mana semua mesin termasuk switching komponen yang menyumbang arus diwakili oleh reaktansi subtransient ( $X''d$ ).
2. Kondisi kedua dimana *short circuit* terjadi pada kondisi  $\frac{1}{2}$  cycle sampai 4 cycle. Kondisi ini disebut periode *transient*, yang hanya generator sinkron dan motor sinkron yang menyumbang arus diwakili oleh reaktansi transient ( $X'd$ ).
3. Kondisi ketiga dimana *short circuit* terjadi pada kondisi 30 cycle. Kondisi ini disebut periode *steady state*, yang mana hanya generator sinkron yang menyumbang arus diwakili oleh reaktansi sinkron ( $X_d$ ). Mesin induksi, motor sinkron, kondensor tidak di pertimbangkan dalam *short circuit* 30 cycle

### 2.2.4 Perhitungan Arus Hubung Singkat

Gangguan yang termasuk dalam hubung singkat simetri adalah gangguan hubung singkat tiga fasa, sedangkan gangguan selain hubung singkat tiga fasa termasuk gangguan hubung singkat asimetri <sup>[2]</sup>. Perhitungan untuk menentukan besar arus hubung singkat dalam sistem tenaga listrik didapatkan dengan cara :

a. Hubung singkat 3 Fasa

Hubung singkat yang terjadi karena adanya melibatkan ketiga fasa. Arus hubung singkat tiga fasa ( $I_{sc_{3\phi}}$ ) diberikan oleh persamaan berikut :

$$I_{sc_{3\phi}} = \frac{V_{LN}}{X_1} \quad (2.1)$$

Keterangan :

$V_{LN}$  : tegangan nominal *line to netral*

$X_1$  : merupakan reaktansi urutan positif

b. Hubung singkat 2 Fasa

Hubung singkat yang terjadi antara dua fasa tanpa terhubung ke tanah. Arus hubung singkat antar fasa ( $I_{sc_{2\phi}}$ ) diberikan oleh persamaan berikut :

$$I_{sc_{2\phi}} = \frac{V_{LL}}{X_1 + X_2} = \frac{\sqrt{3}}{2} I_{sc_3} = 0.866 I_{sc_{3\phi}} \quad (2.2)$$

Keterangan :

$V_{LL}$  : tegangan nominal *line to line*

$X_1$  : reaktansi urutan positif

$X_2$  : reaktansi urutan negative

c. Hubung singkat 1 fasa ke tanah

Hubung singkat ini melibatkan impedansi urutan nol ( $Z_0$ ), dan besarnya arus hubung singkat ini tergantung sistem pentanahan yang digunakan. Arus hubung singkat 1 fasa ( $I_{sc_{1\phi}}$ ) diberikan oleh persamaan berikut :

$$I_{sc_{1\phi}} = \frac{3V_{LN}}{X_1 + X_2 + X_0 + (3xZ_g)} \quad (2.3)$$

Jika sistem menggunakan pentanahan solid maka  $Z_g = 0$ , sehingga persamaan menjadi :

$$I_{sc_{1\phi}} = \frac{3V_{LN}}{X_1 + X_2 + X_0} \quad (2.4)$$

Keterangan :

$X_0$  : reaktansi urutan nol (0)

$Z_g$  : impedansi NGR (Neutral Grounding Resistance)

### 2.3 Sifat-Sifat Gangguan

Bila dilihat dari lamanya waktu terjadi gangguan, maka dapat dikelompokkan menjadi <sup>[1]</sup> :

1. Gangguan yang bersifat temporer, yang dapat hilang dengan sendirinya atau dengan memutuskan sesaat bagian yang terganggu dari sumber tegangannya. Gangguan sementara jika tidak dapat hilang dengan segera, baik hilang dengan sendirinya maupun karena bekerjanya alat pengaman dapat berubah menjadi gangguan permanen.
2. Gangguan yang bersifat permanen, dimana untuk membebaskan gangguan diperlukan tindakan perbaikan dan/atau menyingkirkan penyebab gangguan tersebut.

Saat terjadi gangguan akan mengalir arus yang sangat besar pada fasa yang terganggu menuju titik gangguan, dimana arus gangguan tersebut mempunyai nilai arus yang jauh lebih besar dari rating arus maksimum yang diijinkan, sehingga terjadi kenaikan suhu yang dapat mengakibatkan kerusakan pada peralatan listrik.

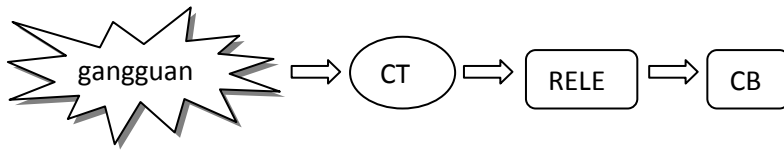
### 2.4 Trafo Arus (*Current Transformer*)

CT atau biasa disebut current transformer adalah suatu peralatan kelistrikan biasanya banyak terdapat pada gardu induk dimana CT berfungsi untuk merubah besaran arus listrik yang digunakan untuk pengukuran perbandingan arus. Misalnya ratio perbandingan arus CT 400 : 5 berarti pada sekunder trafo akan mengeluarkan output 5 Ampere saat sisi primernya dilalui arus sebesar 400 Ampere. CT juga paling banyak digunakan jika arus yang mengalir melalui jaringan terlalu besar mengakibatkan ketidaksesuaian dengan rating peralatan pengukuran maupun proteksi, maka dari itu hasil arus *input* dirubah ke level yang lebih rendah sehingga dapat disesuaikan dengan rating peralatan yang digunakan.

### 2.5 Rele Pengaman pada Sistem Tenaga Listrik

Rele adalah suatu peralatan sistem tenaga listrik yang dirancang untuk memberikan sinyal kepada pemutus tenaga (CB), sehingga CB dapat memutus bila terjadi gangguan atau menghubungkan kembali penyaluran daya. Pemutusan ini dilakukan untuk memisahkan bagian

sistem tenaga listrik yang terjadi gangguan dengan yang tidak terkena gangguan.



**Gambar 2.1** Alur kerja rele saat ada gangguan

Didalam rele terdapat 3 bagian, yaitu *sensing* elemen, *comparison* elemen dan *control* elemen setelah proses itu tercapai lalu dapat mengirimkan sinyal.

Terdapat syarat-syarat yang harus diperhatikan dalam penggunaan rele pengamanan pada sistem proteksi, antara lain <sup>[3]</sup>:

a. Selektifitas

Selektifitas pada sistem proteksi adalah kemampuan rele proteksi untuk melakukan pemilihan *tripping* secara tepat sesuai pada area yang terjadi gangguan. Maksudnya saat terjadi gangguan CB yang seharusnya *Tripping* merupakan CB terdekat dengan gangguan terlebih dahulu sesuai dengan desain sistem proteksi yang dibuat. Dalam kerja rele yang selektifitas ini terdapat 2 kondisi yang mempengaruhi yaitu *time grading*(waktu kerja rele) dan pembagian daerah pengamanan.

b. Sensitifitas

Sensitifitas disini adalah tingkat kepekaan suatu rele untuk dapat mendeteksi terhadap gangguan yang muncul. Suatu rele disebut sensitif apabila parameter operasi utamanya rendah. Artinya, semakin rendah besaran parameter penggerak maka perangkat tersebut dikatakan semakin sensitif. Pada rele-rele numerik, sensitifitas tidak dikaitkan lagi pada perangkat kerasnya tetapi lebih pada aplikasi dan parameter peralatan pengukuran, seperti trafo arus (*current transformer*) atau trafo tegangan (*Voltage transformer*) yang dipakai.

c. Kecepatan

Kecepatan yang dimaksud adalah kecepatan untuk mengisolasi gangguan secepat dan sedini mungkin. Tujuan

utamanya adalah mengamankan kontinuitas pasokan daya dengan cara mengisolasi gangguan.

d. Kehandalan

Rele dapat dikatakan handal dapat dihitung dari jumlah rele yang bekerja saat mengamankan gangguan berbanding dengan jumlah gangguan yang terjadi. Semakin tinggi kehandalan maka sistem tersebut semakin baik dan dapat meminimalkan terjadinya kerusakan akibat gangguan. Terdapat faktor-faktor yang mempengaruhi kehandalan, yaitu *dependable*, *secure*, *availability*.

## 2.6 Rele Arus Lebih

Pada dasarnya rele arus lebih berfungsi sebagai pengaman saat terjadi gangguan hubung singkat, selain itu dapat juga digunakan sebagai pengaman beban lebih (overload). Rele arus lebih merupakan suatu jenis rele yang bekerja berdasarkan besarnya arus masukan, dan apabila besarnya arus masukan melebihi suatu harga tertentu yang dapat diatur ( $I_p$ ) maka rele arus lebih bekerja. Dimana  $I_p$  merupakan arus kerja yang dinyatakan menurut gulungan sekunder dari trafo arus (CT). Bila suatu gangguan terjadi didalam daerah perlindungan rele, besarnya arus gangguan  $I_f$  yang juga dinyatakan terhadap gulungan sekunder CT juga. Rele akan bekerja apabila memenuhi keadaan sebagai berikut <sup>[1]</sup> :

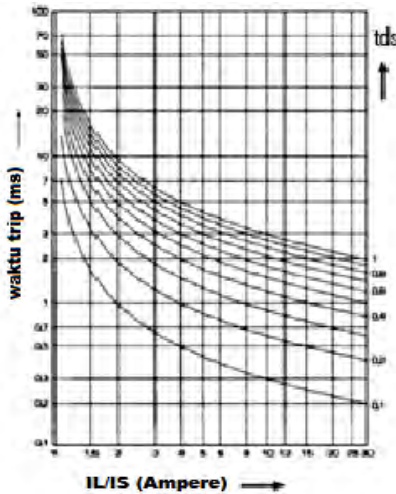
$I_f > I_p \longrightarrow$  rele bekerja, CB *trip*

$I_f < I_p \longrightarrow$  rele tidak bekerja

### 2.6.1 Rele Arus lebih Waktu Inverse

Rele arus lebih waktu invers memiliki waktu operasi yang berbanding terbalik dengan besar nya arus gangguan <sup>[4]</sup>. Jadi, semakin besar arus gangguan maka rele akan bekerja dalam waktu yang semakin cepat. TCC (*time-current characteristic*) merupakan kurva dengan skala dalam *time dial*. Dengan kondisi saat semakin besar setting *time dial*, maka semakin lama waktu operasi dari rele. Karakteristik invers ini dijelaskan dalam standar IEC 60255-3 dan IEEE std 242-2001. Standar-standar ini mendefinisikan jenis karakteristik kurva yang dibedakan oleh

gradien masing-masing, yaitu *normal inverse*, *very inverse*, *long time inverse* dan *extremely inverse*. Kurva inverse ini sering dijumpai dengan *inverse definite minimum time* (IDMT). Kurva IDMT adalah kurva gabungan antara inverse dan definite yang mana seiring dengan arus yang bertambah besar, waktu operasi turun semakin cepat seolah mendekati waktu *definite* minimumnya.



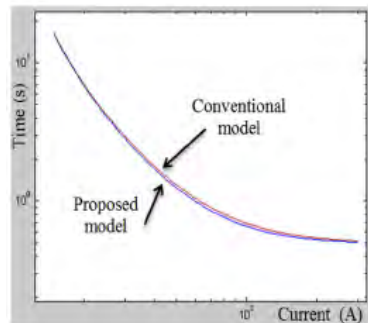
$$tcb = \frac{0,14 \times tD}{\left(\frac{IL}{IS}\right)^{0,02} - 1} \quad (2.5)$$

Gambar 2.2 Karakteristik Kurva Inverse pada Standart IEC

## 2.7 Rele Karakteristik *Tripping* Non-Standart

Dalam kasus tertentu di sistem tenaga industri, rele proteksi umumnya bekerja dibawah prinsip arus lebih untuk menampilkan respon dinamis yang mirip dengan gangguan pada sistem. Pada sistem industri memiliki keragaman dalam sistem proteksi yang terutama proteksi arus lebih. Kurva waktu rele arus lebih lebih sesuai untuk perlindungan peralatan karena dapat mengijinkan kondisi kelebihan arus sementara. Tetapi saat ini komplektivitas koordinasi proteksi meningkat dengan banyaknya rele yang akan dikoordinasikan, terkadang dengan kondisi banyaknya kurva yang dikoordinasikan salah satu kurva akan berhimpitan dengan kurva yang lainnya, sedangkan untuk koordinasi yang baik semua kurva harus mengikuti *grading time* yang sudah ditentukan. Kondisi tersebut mengharuskan sistem koordinasi tidak

harus mengikuti rumusan kurva standart yang dimana akan sulit jika koordinasi melibatkan banyak kurva rele. Tanpa mengikuti rumusan kurva standart, cara yang memungkinkan untuk dilakukan dalam pengoordinasian sistem proteksi adalah dengan memodelkan kurva rele yang dapat disebut kurva non-standart. Berikut gambaran dalam permodelan kurva rele non-standart



**Gambar 2.3.** Kurva rele standart dan kurva rele yang diinginkan

Hasil yang didapatkan adalah pengurangan waktu tripping maksimum dan waktu tripping efektif yang berhubungan dengan *Definite Time Overcurrent* dan *IDMT grading*. Penggunaan kurva ini juga dapat menurunkan *thermal stress* secara drastis yang terjadi pada peralatan sistem tenaga seperti trafo dan kabel.<sup>[12]</sup>

## 2.8 Adaptive Neuro Fuzzy Inference System (ANFIS)

### 2.8.1 Gambaran umum ANFIS

Model Fuzzy dapat digunakan sebagai pengganti dari banyak lapisan. Dalam hal ini sistem dapat dibagi menjadi dua grup, yaitu satu grup berupa jaringan syaraf dengan bobot-bobot fuzzy dan fungsi aktivasi fuzzy dan grup kedua berupa jaringan syaraf dengan input yang di fuzzykan pada lapisan pertama atau kedua, namun bobot-bobot pada jaringan syaraf tersebut tidak di fuzzykan. Menurut Osowski (2004) dan Kusumadewi (2006), Neuro Fuzzy termasuk kelompok kedua. ANFIS (Adaptive Neuro Fuzzy Inference System atau Adaptive Network-based Fuzzy Inference System) adalah arsitektur yang secara fungsional sama dengan fuzzy rule base model Sugeno. Arsitektur ANFIS juga sama



dengan jaringan syaraf dengan fungsi radial dengan sedikit batasan tertentu. Bisa dikatakan bahwa ANFIS adalah suatu metode yang dimana dalam melakukan penyetalan aturan digunakan algortima pembelajaran terhadap sekumpulan data. Pada ANFIS juga memungkinkan aturan-aturan untuk beradaptasi. Agar jaringan dengan fungsi basis radial ekuivalen dengan dengan fuzzy berbasis aturan model sugeno orde 1 ini, diperlukan batasan :

- Keduanya harus memiliki metode agregasi yang sama (rata-rata terbobot atau penjumlahan terbobot) untuk menurunkan semua outputnya.
- Jumlah fungsi aktivasi harus sama dengan jumlah aturan fuzzy (if-then)
- Jika ada beberapa input pada basis aturannya, maka tiap fungsi aktivasi harus sama dengan fungsi keanggotaan tiap-tiap inputnya.
- Fungsi aktivasi dan aturan-aturan fuzzy harus memiliki fungsi yang sama untuk neuron-neuron dan aturan-aturan yang ada disisi outputnya.<sup>[6]</sup>

## 2.10.2 Arsitektur ANFIS

Misalkan input terdiri atas  $Z_{1,t}$  dan  $Z_{2,t}$  dan sebuah output  $Z_t$  dengan aturan model sugeno orde 1. Orde satu dipilih dengan pertimbangan kesederhanaan dan kemudahan perhitungan. Model sugeno orde satu dengan dua aturan fuzzy if-then adalah sebagai berikut:

**Aturan 1 :** *If  $Z_{1,t}$  is  $A_1$  and  $Z_{2,t}$  is  $B_1$  then  $f_1 = p_1 \cdot Z_{1,t} + q_1 \cdot Z_{2,t} + r_1$*



**Premis**



**Konsekuen**

**Aturan 2 :** *If  $Z_{1,t}$  is  $A_2$  and  $Z_{2,t}$  is  $B_2$  then  $f_2 = p_2 \cdot Z_{1,t} + q_2 \cdot Z_{2,t} + r_2$*



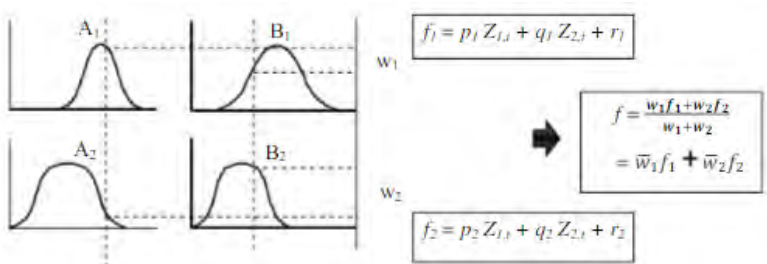
**Premis**



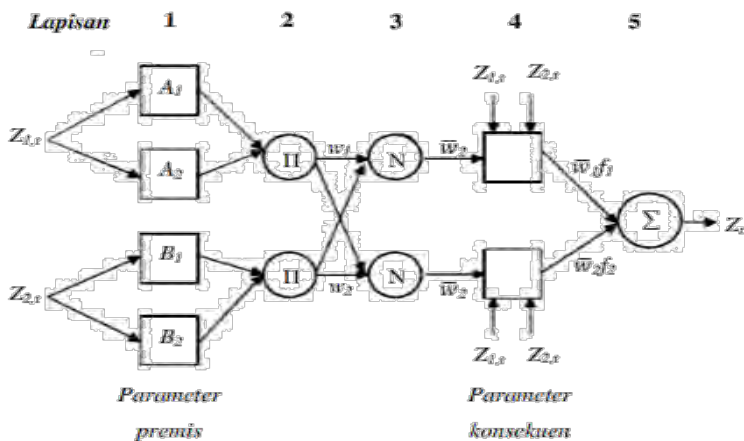
**Konsekuen**

**Gambar 2.4** Aturan arsitektur ANFIS

Dengan  $A_i$  dan  $B_i$  adalah nilai-nilai keanggotaan dan merupakan label linguistik (seperti “kecil” atau “besar”),  $p_i$ ,  $q_i$  dan  $r_i$  adalah parameter konsekuen.



**Gambar 2.5** ANFIS dengan model sugeno



**Gambar 2.6** Arsitektur jaringan ANFIS<sup>[7]</sup>

### Lapisan 1:

Lapisan ini merupakan lapisan fuzzifikasi. Pada lapisan ini tiap neuron adaptif terhadap parameter suatu aktivasi. Output dari tiap neuron berupa derajat keanggotaan yang diberikan oleh fungsi keanggotaan input. Misalkan fungsi keanggotaan Generalized Bell diberikan sebagai :

$$\mu(Z) = \frac{1}{1 + \left| \frac{Z-c}{a} \right|^{2b}} \quad (2.6)$$

Dengan  $Z$  adalah input, dalam hal ini  $Z = \{Z_{1,t}, Z_{2,t}\}$  dan  $\{a, b, \text{ dan } c\}$  adalah parameter-parameter, biasanya  $b=1$ . Jika nilai parameter-parameter ini berubah, maka bentuk kurva yang terjadi akan ikut berubah. Parameter-parameter ini biasanya disebut dengan nama **parameter premis**.

#### Lapisan 2:

Lapisan ini berupa neuron tetap (diberi symbol  $\Pi$ ) merupakan hasil kali dari semua masukan, sebagai berikut :

$$w_1 = \mu_{Ai} \cdot \mu_{Bi} \quad (2.7)$$

Biasanya digunakan operator AND. Hasil perhitungan ini disebut firing strength dari sebuah aturan. Tiap neuron mempresentasikan aturan ke- $i$ .

#### Lapisan 3:

Tiap neuron pada lapisan ini berupa neuron tetap (diberi symbol  $N$ ) Merupakan hasil perhitungan rasio dari firing strength ke- $i$  ( $w_i$ ) terhadap jumlah dari keseluruhan firing strength pada lapisan kedua, sebagai berikut :

$$w_1 = \frac{w_1}{w_1 + w_2}, i = 1, 2 \quad (2.8)$$

Hasil perhitungan ini disebut normalized firing strength.

#### Lapisan 4:

Lapisan ini berupa neuron yang merupakan neuron adaptif terhadap suatu output, sebagai berikut :

$$w_i f_i = w_i (p_i Z_{1,t} + q_i Z_{2,t} + r_i) \quad (2.9)$$

Dengan  $w_i$  adalah normalized firing strength pada lapisan ketiga dan  $p_i, q_i$  dan  $r_i$  adalah parameter-parameter pada neuron tersebut. Parameter-parameter ini biasa disebut **parameter konsekuen**.

#### Lapisan 5:

Lapisan ini berupa neuron tunggal (diberi symbol  $\Sigma$ ) merupakan hasil penjumlahan seluruh output dari lapisan keempat, sebagai berikut :

$$\sum_i w_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \quad (2.10)$$

### 2.8.3 Algoritma Pembelajaran Hybrid

Pada saat parameter premis ditemukan, keluaran keseluruhan merupakan kombinasi linier dari konsekuen parameter, yaitu :

$$\begin{aligned} f &= \frac{w_1}{w_1 + w_2} f_1 + \frac{w_2}{w_1 + w_2} f_2 \\ &= w_1 (p_1 Z_{1,t} + q_1 Z_{2,t} + r_1) + w_2 (p_2 Z_{1,t} + q_2 Z_{2,t} + r_2) \\ &= (w_1 Z_{1,t}) p_1 + (w_1 Z_{2,t}) q_1 + (w_1) r_1 + (w_2 Z_{1,t}) p_2 + \\ &\quad (w_2 Z_{2,t}) q_2 + (w_2) r_2 \end{aligned} \quad (2.11)$$

Algoritma hibrida akan mengatur parameter-parameter konsekuen  $p_i$ ,  $q_i$ ,  $r_i$  secara maju (forward) dan akan mengatur parameter-parameter a, b dan c secara mundur (backward). Pada langkah maju, input jaringan akan merambat maju sampai lapisan keempat. Parameter-parameter konsekuen akan diidentifikasi dengan menggunakan least-square. Sedangkan pada langkah mundur, error sinyal akan merambat mundur dan parameter-parameter premis akan diperbaiki dengan metode gradient descent.<sup>[7]</sup>

Untuk memberi nilai awal pada parameter premis (pada bagian membership function), biasanya digunakan Fuzzy Clustering Mean (FCM). Pada arsitektur ANFIS, node yang bersifat adaptatif terdapat pada layer 1 dan 4. Node pada layer 1 mengandung parameter premis yang nonlinier sedangkan pada layer 4 mengandung parameter konsekuen yang linier. Untuk memperbarui parameter – parameter tersebut (dengan kata lain proses belajar dari jaringan saraf), kita memerlukan proses learning atau training. ANFIS menggunakan hybrid supervised method yang berbasis pada dua metode, yaitu least – squares dan gradient descent.

Pada arah maju (forward), parameter premis dibuat tetap. Dengan menggunakan metode Least Square Estimator (LSE), parameter konsekuen diperbaiki berdasarkan pasangan data pada training set. Metode LSE dapat diterapkan karena parameter konsekuen yang diperbaiki bersifat linier. Setelah parameter konsekuen diperoleh, data masukan dilewatkan jaringan adaptif kembali dan hasil keluaran jaringan adaptif ini dibandingkan dengan keluaran yang diharapkan (target). Pada arah mundur (backward), parameter konsekuen dibuat

tetap. Kesalahan (error) antara keluaran jaringan adaptif dan target dipropagasikan balik menggunakan gradient descent untuk memperbarui parameter premis. Satu tahap pembelajaran maju – mundur ini dinamakan satu epoch.

## BAB III PERANCANGAN SOFTWARE DAN HARDWARE

Dalam pembahasan Bab III ini akan dilakukan dan dijelaskan bagaimana perancangan software dan hardware dalam tugas akhir ini. Pertama-tama dalam perancangan ini membuat perancangan software menggunakan Matlab 2013 dengan interface GUI Matlab.

### 3.1 Perancangan Software

Dalam perancangan software yang menggunakan Matlab 2013 dengan interface GUI bertujuan sebagai simulasi hasil dari pembelajaran metode ANFIS dan hasil permodelan kurva Non-standart. Sebelum melakukan perancangan software simulasi terlebih dahulu melihat data-data rele yang terdapat pada single line diagram PT.HESS Indonesia.

#### 3.1.1 Pegambilan Data Rele Pada Etap 12.6

Untuk single line diagram dari PT. HESS diambil beberapa bagian yang tipikal agar memudahkan dalam pengamatan dan menganalisa bentuk kurva koordinasi rele. Data-data rele tersebut meliputi sebagai berikut :

**Tabel 3.1** Data rele PT.HESS Indonesia

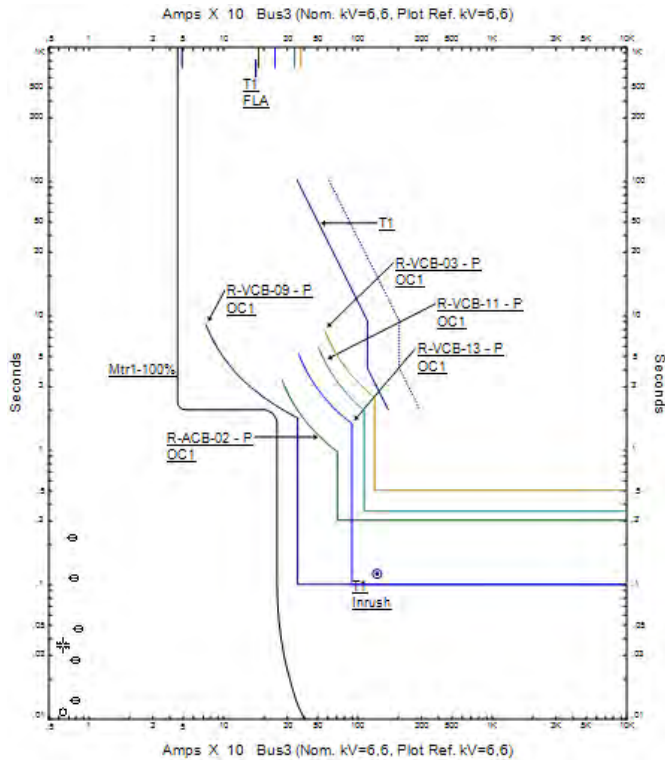
ID	CT Ratio	Setting	
R-VCB-09	60/5	Curve	IEC- Normal Inverse
		Current setting	1,65
		Time dial	0,65
		Current high set	11,95
		Time delay	0,1
R-ACB-02	350/5	Curve	IEC- Normal Inverse
		Current setting	1,15
		Time dial	0,15
		Current high set	3,75
		Time delay	0,3
R-VCB-13	180/5	Curve	IEC- Normal Inverse
		Current	1,35

		setting	
		Time dial	0,3
		Current high set	5,05
		Time delay	0,1
<b>R-VCB-11</b>	400/5	Curve	IEC- Normal Inverse
		Current setting	0,85
		Time dial	0,34
		Current high set	2,8
		Time delay	0,35
<b>R-VCB-03</b>	400/1	Curve	IEC- Normal Inverse
		Current setting	0,95
		Time dial	0,45
		Current high set	3,36
		Time delay	0,5

Setelah didapatkan data rele selanjutnya mengamati kurva rele dengan menggunakan software Etap 2013 untuk mengondisikan atau menyimulasikan adanya salah satu dari beberapa kurva rele agar terjadi crossover(berhimpitan) antar kurva rele. Cara untuk mengondisikan agar kurva rele dapat berhimpitan dengan mensetting nilai tds sampai terjadinya cross antar kurva. Sebagai contoh kurva rele R-VCB-9 dikondisikan berhimpitan dengan kurva rele R-ACB-02, dengan kondisi saling berhimpitan akan dilakukan permodelan kurva non-standart dengan kondisi mengubah nilai tds sudah tidak bisa dilakukan. Berikut gambar single line diagram PT.HESS Indonesia dengan mengambil beberapa bagian yang tipikal :







**Gambar 3.2** Kurva rele R-VCB-9 saat kondisi normal

Pada gambar diatas menunjukkan kurva rele R-VCB-9 saat kondisi setting dalam keadaan normal. Maksudnya setting dalam keadaan normal kurva dapat bekerja secara normal tanpa adanya trip secara bersamaan dengan kurva lainnya dalam kasus ini dengan kurva R-ACB-02. Kondisi normal pada kurva rele R-VCB-9 dengan settingan :

**Tabel 3.2** Kondisi normal setting rele R-VCB-9

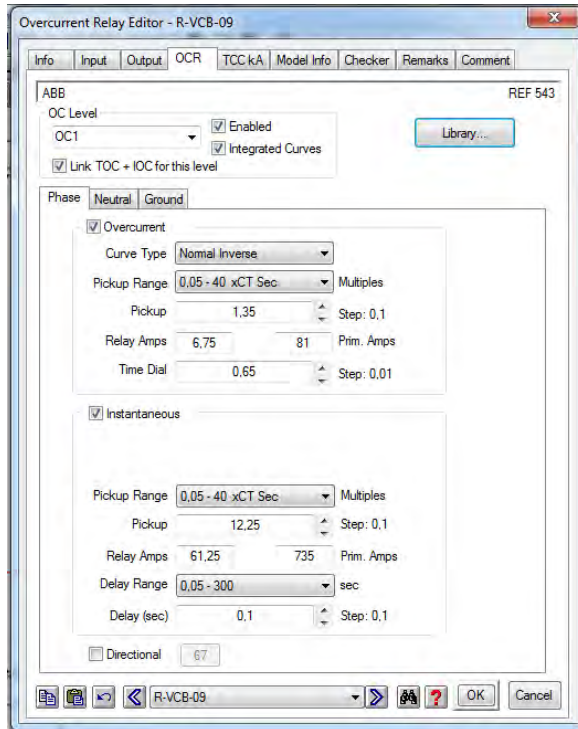
ID	CT Ratio	Setting	
		Curve	IEC- Normal Inverse
R-VCB-09	60/5	Current setting	1,65
		Time dial	0,5
		Current high set	11,95
		Time delay	0,1

Dimana saat dikondisikan berhimpitan dengan kurva rele R-ACB-2 setting pada kurva rele R-VCB-9 akan disetting ulang mulai nilai current setting sampai current high set sehingga kurva R-VCB-9 dapat berhimpitan dengan kurva R-ACB-2. Berikut merupakan setting awal dari kurva rele R-ACB-02 :

**Tabel 3.3** Kondisi normal setting rele R-ACB-02

ID	CT Ratio	Setting	
		Curve	IEC- Normal Inverse
R-ACB-02	350/5	Current setting	1,15
		Time dial	0,15
		Current high set	3,75
		Time delay	0,3

Selanjutnya dilakukan setting ulang pada kurva rele R-VCB-9 dimana settingan akan dikondisikan berhimpitan dengan kurva rele R-ACB-02. Berikut merupakan setting ulang rele R-VCB-9 pada Etap 12.6 :

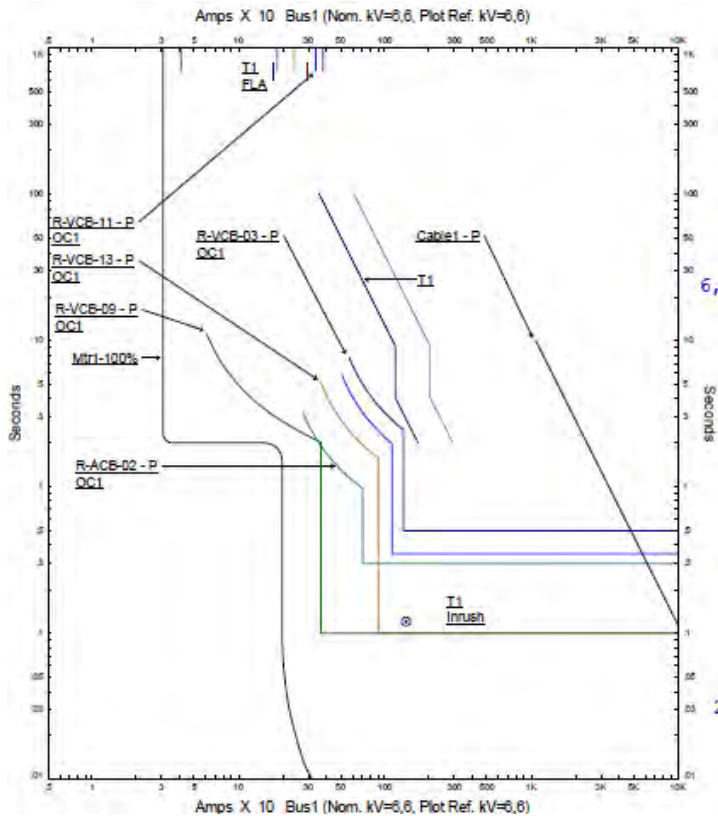


**Gambar 3.3** Setting ulang rele R-VCB-9

**Tabel 3.4** Setting rele R-VCB-9 saat kondisi berhimpitan

ID	CT Ratio	Setting	
R-VCB-09	60/5	Curve	IEC- Normal Inverse
		Current setting	1,35
		Time dial	0,65
		Current high set	12,25
		Time delay	0,1

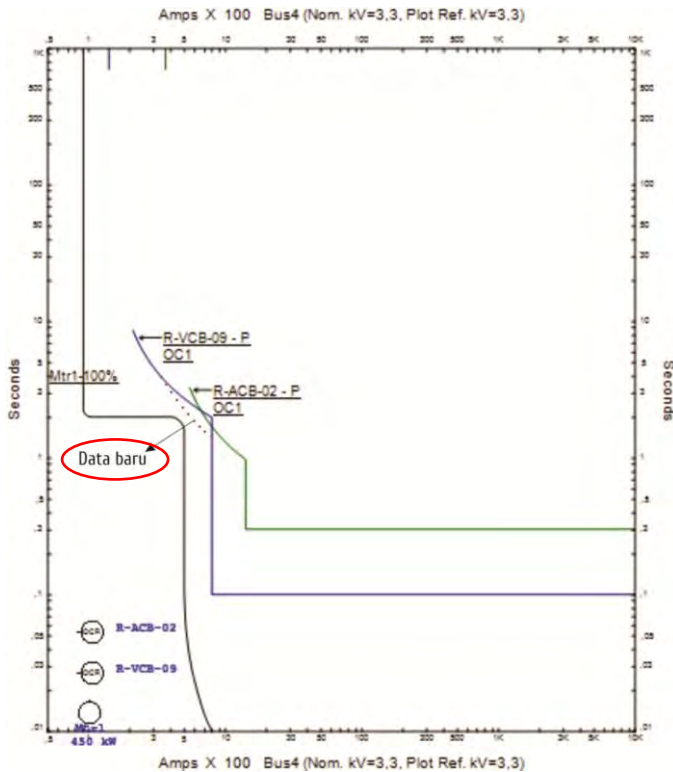
Setting ulang pada rele R-VCB-9 dilakukan dengan menurunkan nilai current setting (pick-up) dari nilai 1,65 menjadi 1,35 dan menaikkan nilai pada current high set (pick-up) dari nilai 11,95 menjadi 12,25. Sehingga pada plot kurva akan berubah seperti gambar dibawah ini :



**Gambar 3.4** Kurva rele R-VCB-9 kondisi berhimpitan

Pada gambar diatas plot kurva pada Etap 12.6 sudah menunjukkan pengondisian kurva rele R-VCB-9 berhimpitan dengan kurva rele R-ACB-02. Dimana pada kasus ini kurva rele R-VCB-9 sudah tidak bisa disetting lagi dengan kondisi saat disetting menurunkan lowest dan highest kurva akan bersinggungan dengan kurva rele motor dibelakangnya, yang akan mengakibatkan trip pada rele R-VCB-9 saat motor starting. Jika disetting dengan menaikkan nilai lowest dan highsetnya kurva rele R-VCB-9 akan semakin bersinggungan dengan kurva R-ACB-2. Dari kasus tersebut diperlukan permodelan kurva baru sebagai solusi agar antar kurva tidak bersinggungan lagi.

Permodelan kurva non-standart dilakukan dengan mengambil beberapa data baru dimana akan membentuk suatu kurva baru yang sudah tidak bersinggungan lagi dengan kurva lainnya. Berikut merupakan contoh dari pengambilan data baru :

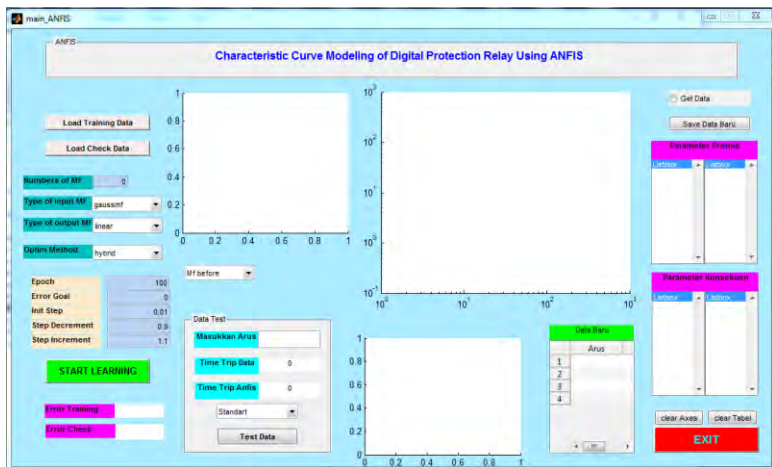


**Gambar 3.5** Kurva rele dengan titik data baru

Dengan menggunakan titik baru seperti diperlihatkan titik “data baru” pada gambar diatas didapatkan kurva rele yang baru atau sudah non-standart dengan hasil kurva tidak bersinggungan dan dapat bekerja sesuai dengan rencana koordinasi proteksi.

### 3.1.2 Perancangan Interface GUI Matlab 2013

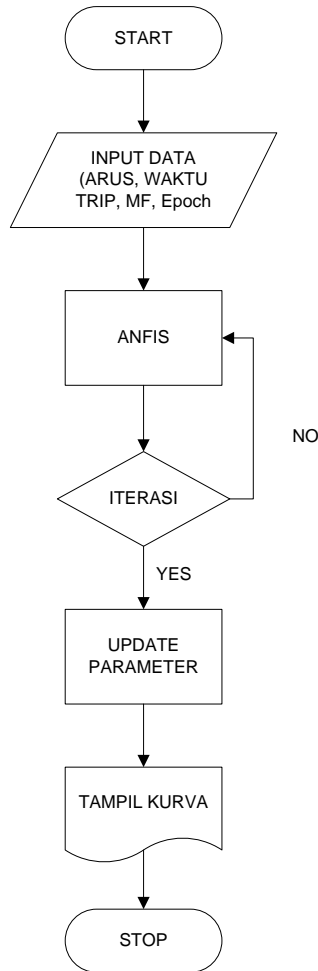
Dalam perancangan interface tugas akhir ini bertujuan agar user dapat membuktikan secara langsung hasil dari pembelajaran metode yang digunakan dalam permodelan kurva rele sudah mencapai hasil yang diinginkan atau belum. Perancangan interface menggunakan software Matlab 2013 dengan program GUI Matlab 2013. Dalam GUI akan dirancang interface yang akan menampilkan pengambilan data, input data ANFIS, hasil parameter ANFIS (parameter premis dan konsekuen),serta hasil learning ANFIS berupa plot kurva dan kolom untuk membuktikan hasil pembelajaran metode ANFIS dalam memodelkan kurva rele. Berikut merupakan gambar dari interface GUI yang dibuat :



Gambar 3.6 Interface GUI Matlab 2013

Dalam program GUI sudah terdapat program yang mengolah pembelajaran ANFIS dengan memanfaatkan script program dari ANFIS yang tersedia di Matlab 2013. Dengan begitu dalam interface GUI bisa langsung mengolah data dan melearning data sehingga menghasilkan kurva baru atau non-standart. Flowchart permodelan kurva non-standart dengan ANFIS adalah sebagai berikut :





**Gambar 3.7** Flowchart permodelan kurva non-standart

Proses training ANFIS dengan diawali inisialisasi nilai awal parameter. Inisialisasi menggunakan FCM (fuzzy clustering mean) yang merupakan proses penglompokan data dimana hasil akhir didapat parameter premis dari data yaitu perhitungan mean dan standar deviasai. Setelah didapatkan parameter premis dari proses FCM, berikutnya masuk ke struktur ANFIS dimana pada bagaian ini data akan diolah pada tiap-tiap

layernya yang ada pada proses pembelajaran ANFIS. Pada layer 5 data akan menghasilkan nilai output. Pada bagian struktur ANFIS ini data akan melakukan iterasi hingga di dapat nilai error yang kecil menunjukkan hasil sudah mendekati target. Setelah output mencapai target maka akan diperoleh parameter terbaru dan tampilan sebuah kurva ANFIS. Parameter terbaru ini adalah update dari parameter lama yang berupa premis dan konsekuen.

### **3.2 Perancangan Hardware dan Integrasi**

#### **3.2.1 Perancangan Hardware**

Dalam perancangan hardware untuk pengolahan data menggunakan mikrokontroller ARM dan interface pada hardware menggunakan LCD grafik 128x64. Dalam menggunakan mikrokontroller ARM dimaksudkan agar lebih kompatibel dengan berbagai bahasa pemrograman. Untuk tampilan lcd grafik 128x64 dapat menampilkan arus serta waktu trip secara langsung (*real time*). Berikut merupakan tampilan dari mikrokontroller ARM :



**Gambar 3.8** Mikrokontroller ARM

Berikutnya merupakan tampilan dari LCD grafik 128x64 yang digunakan untuk menampilkan data dan hasil dari program yang sudah dibuat dan menunjukkan bahwa hasil sudah sesuai atau belum dengan hasil yang dihasilkan software.



**Gambar 3.9** LCD 128x64

## **BAB IV**

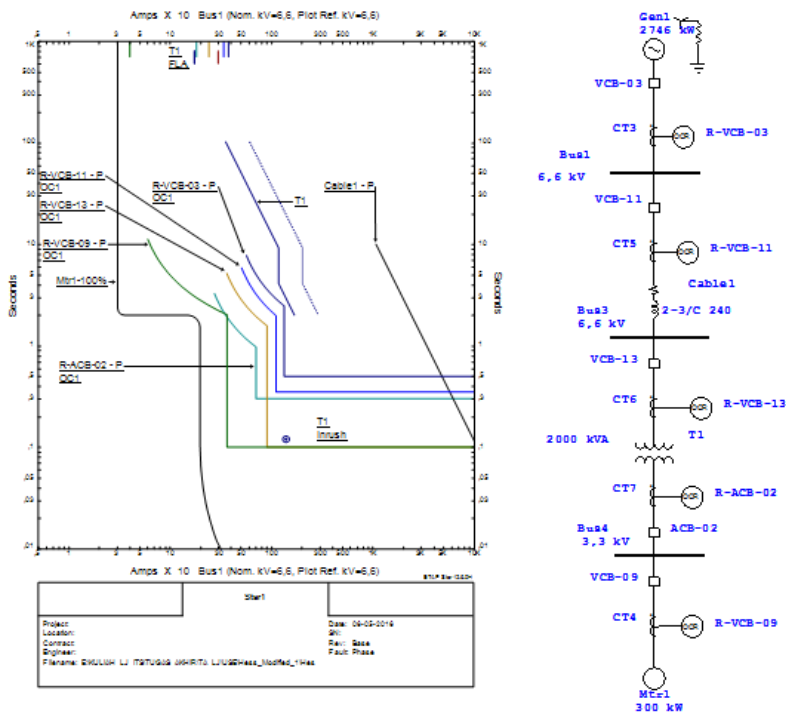
### **HASIL PENGUJIAN DAN ANALISA DATA**

Dalam tugas akhir ini membahas tentang permodelan kurva rele dengan menggunakan single line diagram PT.HESS Indonesia yang telah di ambil salah satu tipikal koordinasi proteksinya untuk memudahkan pemahaman tentang permodelan kurva non-standart. Untuk melakukan permodelan kurva dilakukan terlebih dahulu pengambilan data rele dan merencanakan kurva rele mana yang akan dilakukan permodelan nantinya, dalam kasus ini dipilih kurva rele dengan name code R-VCB-9. Kurva rele R-VCB-9 nantinya akan dipilih setting relenya dengan maksud agar kurva rele R-VCB-9 dapat bersinggungan dengan kurva rele R-ACB-2 yang dimana akan dijadikan permasalahan dalam tugas akhir ini. Selanjutnya dilakukan pengambilan data lowset dan highest pada rele R-VCB-9 yang sudah dipilih setting relenya dan mencari waktu trip dengan memasukkan kedalam rumus normal inverse yang dilakukan dengan software Microsoft excel dan dijadikan sebuah data pengujian. Setelah didapatkan data berupa arus dan waktu trip yang masih data kurva normal inverse, langkah selanjutnya membuat program interface GUI Matlab 2013 yang didalamnya sudah termasuk program metode pembelajaran ANFIS sebagai metode dalam tugas akhir ini.

Setelah interface selesai dibuat dan data rele R-VCB-9 yang masih normal inverse (standart IEC) didapatkan, selanjutnya akan dilakukan pengujian dengan metode *Adaptive Neuro Fuzzy Inference System* (ANFIS) terlebih dahulu dengan data rele R-VCB-9 yang normal inverse (standart IEC). Setelah dilakukan pengujian pada data rele standart IEC pada rele R-VCB-9 akan dilakukan pengambilan data baru berupa titik-titik data baru pada plot kurva interface GUI. Data baru yang sudah didapat akan digabungkan dengan data yang lama sehingga membentuk data non-standart. Setelah didapatkan data non-standart dilakukan pengujian dengan ANFIS sehingga didapatkan data kurva non-standart hasil *learning* ANFIS. Hasil kurva non-standart akan diuji juga pada Etap 12.6 dan dilakukan pembuktian dengan plot pada Etap 12.6. Pada tugas akhir ini data non-standart hasil learning ANFIS juga diujikan langsung pada *hardware Digital Relay Protection* (DPR).

## 4.1 Pengambilan Data Rele

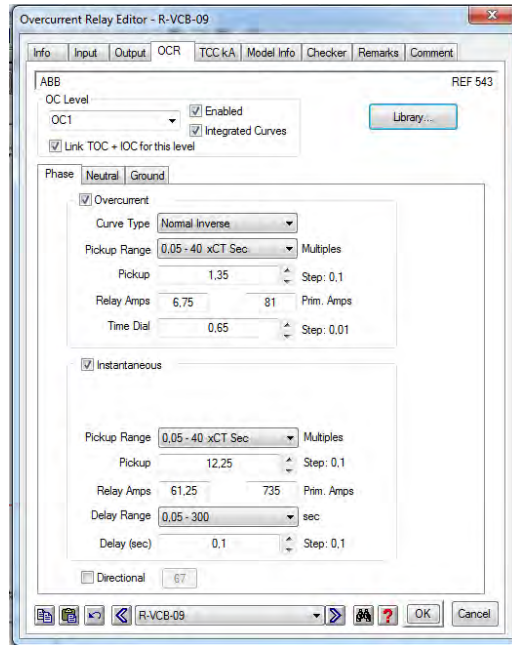
Pengambilan data rele dilakukan pada software Etap 12.6 dengan menggunakan single line diagram PT.HESS Indonesia yang sudah di modifikasi dengan mengambil salah satu tipikal settingan rele. Berikut ini merupakan contoh gambar single line diagram yang digunakan :



**Gambar 4.1** Single line diagram dan koordinasi rele

Dengan mengondisikan kurva pada rele R-VCB-9 bersinggungan dengan kurva rele R-ACB-02 yang artinya pada arus tertentu kedua rele akan mentriptikan CB (*Circuit Breaker*) secara bersama. Dalam hal ini diperlukan permodelan kurva rele pada R-VCB-9 dengan syarat pergantian setting nilai lowset dan highset sudah tidak dapat dilakukan lagi dan perubahan jenis kurva rele juga tidak bisa dilakukan. Dengan

software Etap 12.6 akan dilihat data pada rele R-VCB-9, berikut ini merupakan setting pada rele R-VCB-9 :



**Gambar 4.2** Setting rele R-VCB-9 di Etap 12.6

Terlihat pada gambar diatas setting tap lowset pada 1,35 dengan CT 60/5 didapat  $I_{Set}$  pada rele R-VCB-9 bernilai 81 A dan setting highset pada nilai 735 A. Dari data tersebut akan dibuat data arus dan waktu trip menggunakan software microsoft excel, dengan melihat rele R-VCB-9 menggunakan tipe kurva normal invers maka untuk mencari waktu trip menggunakan rumus sebagai berikut :

$$tcb = \frac{0,14 \times D}{\left(\frac{I}{I_s}\right)^{0,02} - 1} \quad (4.1)$$

Dengan menggunakan rumus kurva normal invers dapat diketahui waktu trip pada rele R-VCB-9. Dengan mengambil nilai lowset diatas nilai  $I_{set}$

(81 A) yaitu 95 A didapatkan waktu trip saat lowset adalah sebagai berikut :

$$t = \frac{0,14 \times 0.65}{(95/81)^{0,02} - 1}$$

$$t = 28.494 \text{ detik} \quad (4.2)$$

Untuk langkah berikutnya dilakukan pengambilan data waktu trip dari setting arus lowset sampai highsetnya menggunakan software microsoft excel dengan berpedoman pada rumus kurva normal inverse, berikut merupakan tabel data normal inverse pada microsoft excel :

**Tabel 4.1** Data kurva normal inverse (standart IEC)

No.	Arus (A)	Trip (S)
1	95	28,494
2	145	7,769
3	195	5,134
4	245	4,066
5	295	3,475
6	345	3,095
7	395	2,826
8	445	2,626
9	495	2,468
10	545	2,342
11	595	2,237
12	645	2,148
13	695	2,072
14	735	2,018

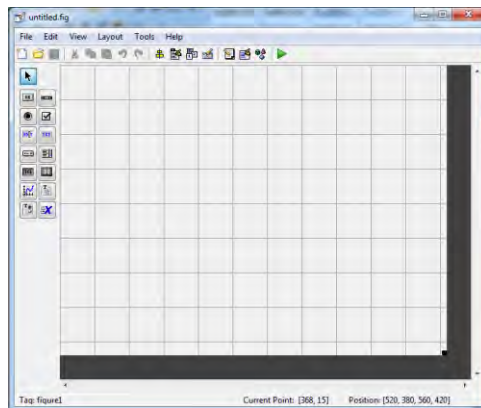
Data diatas merupakan data rele R-VCB-9 dengan jarak antar data arus sebesar 50 A dan berjumlah 14 data inputan. Data tersebut juga dapat diubah-ubah dengan jarak antar arus sebesar 10 A, 15 A, 20 A, 25 A sampai 75 A yang pastinya jumlah data yang didapatkan berbeda-

beda. Data yang didapatkan akan di learning menggunakan metode ANFIS dengan interface GUI Matlab 2013.

## 4.2 Pembuatan Interface GUI Matlab

Dalam tugas akhir ini digunakan interface menggunakan GUI Matlab 2013 sebagai media untuk melakukan proses learning data dan pembuktian dari pembelajaran metode yang digunakan. Untuk pembuatan GUI Matlab dapat mengikuti langkah sebagai berikut :

1. Buka Matlab 2013 dan setelah itu ketik pada command window *"guide"*.
2. Lalu akan muncul tampilan *"GUIDE quick start"* , pilih menu *"create new GUI"*.
3. Setelah itu akan muncul menu *"GUIDE"* template dan pilih *"Blank GUI default"*.
4. Pada akhirnya akan muncul workspace untuk pembuatan arsitektur GUI seperti pada gambar dibawah ini :



**Gambar 4.3** Workspace GUI Matlab 2013

Untuk tugas akhir ini sudah dibuat interface pada GUI Matlab dimana akan ditunjukkan pada gambar dibawah ini :





Gambar 4.4 Tampilan interface GUI

Dengan melihat tampilan interface pada dapat dilakukan proses training dengan mengambil data terlebih dahulu dengan "load training data" dan mengecek data dengan data yang sama melalui "load check data". Dalam kolom "Number of MF" berguna untuk menginputkan jumlah MF (*membership function*) sesuai dengan yang diinginkan user. Untuk tipe input dan output MF dipilih sesuai dengan keinginan user untuk menentukan jenis membership function dimana dalam tugas akhir ini ditetapkan MF jenis *GAUSS* dengan output linier. Dan kolom epoch berfungsi untuk menentukan jumlah iterasi yang diinginkan. Dalam kolom data test terdapat kolom masukkan arus yang berfungsi untuk menguji hasil waktu trip dari pembelajaran ANFIS dan hasil dari perhitungan rumus. Dalam interface GUI tersebut masih banyak fungsi yang bisa digunakan antara lain menyimpan hasil pembelajaran ANFIS, tombol penentuan titik data baru, kolom tampilan data baru, kolom parameter premis dan parameter konsekuen, dll.

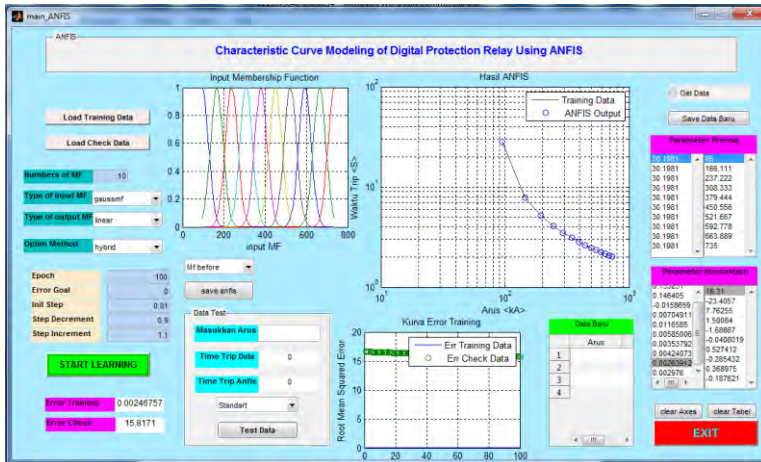
### 4.3 Pengujian Data Normal Inverse (Standart IEC)

Dalam percobaan ini akan dilakukan dengan memasukkan data yang sudah ada untuk dilearning menggunakan metoda ANFIS dengan melihat hasil waktu trip CB apakah sudah mendekati target dan rata-rata error. Berikut ini data yang akan yang akan diujikan dengan menggunakan jarak nilai antara arusnya 50 A :

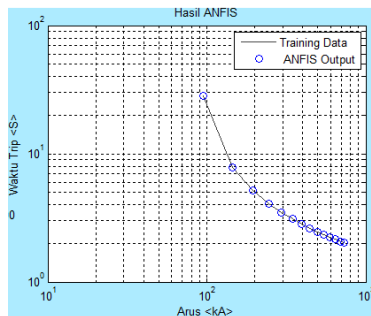
**Tabel 4.2** Data dengan nilai jarak antara arus sebesar 50 A

<b>No.</b>	<b>Arus (A)</b>	<b>Trip (S)</b>
<b>1</b>	95	28,494
<b>2</b>	145	7,769
<b>3</b>	195	5,134
<b>4</b>	245	4,066
<b>5</b>	295	3,475
<b>6</b>	345	3,095
<b>7</b>	395	2,826
<b>8</b>	445	2,626
<b>9</b>	495	2,468
<b>10</b>	545	2,342
<b>11</b>	595	2,237
<b>12</b>	645	2,148
<b>13</b>	695	2,072
<b>14</b>	735	2,018

Dengan menggunakan data seperti pada tabel 4.2 akan dilakukan learning dengan metode ANFIS yang ditampilkan pada interface GUI dan mencari nilai MF yang sesuai dengan jumlah data yang ada. Berikut ini merupakan hasil pengujian data normal inverse (standart IEC) dengan tampilan interface GUI :



**Gambar 4.5** Interface GUI percobaan data jarak antar arus 50 A



**Gambar 4.6** Plot kurva hasil training

Melihat pada gambar 4.5 dan gambar 4.6 diatas dengan jumlah 14 data yang ditraining dan jumlah MF (Membership Function) yang ditentukan sebesar 10 MF dan jumlah iterasi sebanyak 100 kali didapatkan hasil output ANFIS sudah memenuhi target dengan MSE (mean squared error) sebesar 0,0024%. Selanjutnya dilakukan pengujian data test dengan mengambil beberapa data sesuai pada tabel 4.2 data target (data standart IEC) dan beberapa data diluar data target. Gambar dibawah ini menunjukkan pengujian data test :

Data Test	
Masukkan Arus	145
Time Trip Data	7.76864
Time Trip Anfis	7.76876
	Standart
<b>Test Data</b>	

**Gambar 4.7** Pengujian data test sesuai data training

Data Test	
Masukkan Arus	693.3
Time Trip Data	2.07405
Time Trip Anfis	2.07738
	Standart
Test Data	

**Gambar 4.8** Pengujian data test diluar data training

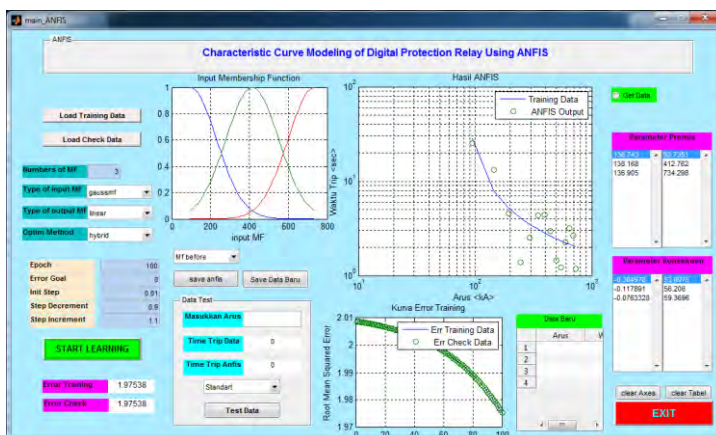
Pengujian data test yang menggunakan nilai sesuai data target hasil waktu trip CB dari perhitungan rumus maupun hasil training dari ANFIS mempunyai nilai yang hampir sama dan error yang dihasilkan sebesar 0,0015%. Sedangkan untuk data test diluar data target dengan mengambil nilai diantara 645 A sampai 695 A dan dipilih arus sebesar 693,3 A yang menghasilkan waktu trip CB dari perhitungan rumus dibandingkan dengan hasil learning ANFIS mempunyai perbedaan waktu 0,003s dengan nilai error 0,33%. Pengujian pada data diluar data training memiliki error dengan nilai 0,33% dikatakan masih kecil karena nilai error masih dibawah 5 %. Secara rincinya dapat dilihat pada tabel dibawah ini :

**Tabel 4.3** Hasil error pengujian data test normal inverse

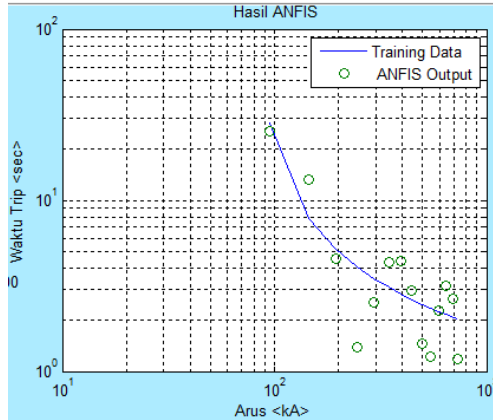
Jenis Data	Arus (A)	Trip Rumus (s)	Trip Anfis (s)	Error (%)
Sesuai data training	145	7,76864	7,76876	0,0015
Diluar data training	693,3	2,07405	2,07738	0,33

#### 4.3.1 Pengujian keakuratan dengan mengganti jumlah Membership Function (MF)

Pengujian kali ini dilakukan untuk membuktikan keakuratan learning ANFIS untuk permodelan kurva data normal inverse dengan mengubah-ubah nilai membership functionnya. Data yang akan dilakukan pengujian merupakan data normal inverse seperti pada tabel 4.2 dengan jumlah data sebanyak 14 data dengan jarak antar data linier sebesar 50A. Percobaan pertama dilakukan pada nilai membership function (MF) = 3 dan dengan nilai epoch konstan 100, dengan melihat hasil plot kurva pada interface GUI dan test data dengan perbandingan hasil waktu trip rumus normal inverse dan hasil waktu trip ANFIS. Berikut hasil pengujian pada interface GUI :



**Gambar 4.9** Hasil percobaan MF 3 pada interface GUI



**Gambar 4.10** Hasil plot kurva ANFIS MF = 3

Dengan pengujian nilai MF = 3 pada gambar 4.10 hasil learning ANFIS yang ditunjukkan lingkaran warna hijau masih jauh untuk memenuhi target data sesuai dengan data target yang ditampilkan oleh garis warna biru dengan nilai rata-rata error = 47,03%. Berikut error perbandingan hasil learning ANFIS dengan hasil target data normal inverse :

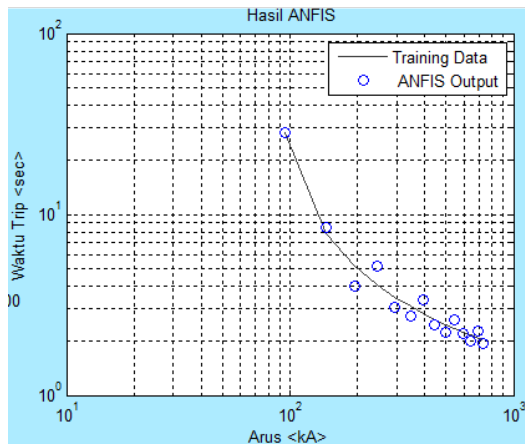
**Tabel 4.4** Hasil error perbandingan trip ANFIS dan trip target dengan MF 3

No.	Arus (A)	Trip Anfis (s)	Trip Target (s)	Error (%)
1	95	25,51003268	28,494	11,69723
2	145	13,14010869	7,769	40,875679
3	195	4,54318089	5,134	13,004525
4	245	1,384043586	4,066	193,77688
5	295	2,533646992	3,475	37,154071
6	345	4,332601738	3,095	28,564863
7	395	4,40793801	2,826	35,888391
8	445	2,970388029	2,626	11,594042
9	495	1,463494972	2,468	68,637409
10	545	1,227996099	2,342	90,717218

**Lanjutan Tabel 4.4** Hasil error perbandingan trip ANFIS dan trip target dengan MF 3

No.	Arus (A)	Trip Anfis (s)	Trip Target (s)	Error (%)
11	595	2,281348242	2,237	1,9439488
12	645	3,156045691	2,148	31,940149
13	695	2,638124586	2,072	21,459357
14	735	1,178182619	2,018	71,280748

Pengujian selanjutnya menambah nilai MF = 5 dengan nilai epoch tetap 100 konstan. Penambahan nilai MF diharapkan dapat memperbaiki learning untuk mencapai hasil target sesuai dengan data normal inverse pada tabel 4.2. Berikut hasil plot pada interface GUI dengan MF = 5 :



**Gambar 4.11** Hasil plot kurva learning ANFIS MF = 5

Pada gambar 4.11 menunjukkan dengan penambahan nilai MF = 5 hasil learning ANFIS (lingkaran warna biru) mulai mendekati kurva data target (garis warna hitam), namun hasil learning ANFIS masih belum mencapai target data normal inverse (standart IEC). Hasil learning ANFIS dengan nilai MF = 5 masih terdapat error, tetapi nilai error dengan penambahan nilai MF dari 3 ke 5 menjadi semakin kecil. Berikut

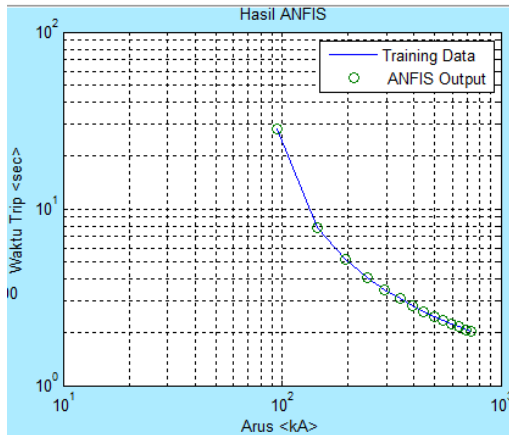
hasil error perbandingan nilai learning ANFIS dengan hasil data normal inverse :

**Tabel 4.5** Hasil error perbandingan trip ANFIS dan trip target dengan MF 5

No.	Arus (A)	Trip ANFIS (s)	Trip Target (s)	Error (%)
1	95	28,27544624	28,494	0,7729454
2	145	8,431386702	7,769	7,8562012
3	195	4,012134595	5,134	27,961809
4	245	5,190220854	4,066	21,660366
5	295	3,077327276	3,475	12,922666
6	345	2,732611054	3,095	13,261637
7	395	3,370642745	2,826	16,158424
8	445	2,469764937	2,626	6,3259082
9	495	2,214369684	2,468	11,453838
10	545	2,624552356	2,342	10,765735
11	595	2,192977093	2,237	2,0074495
12	645	1,990504649	2,148	7,9123328
13	695	2,247756269	2,072	7,8191871
14	735	1,93746107	2,018	4,1569315

Pada tabel 4.5 dengan penambahan MF=5 menunjukkan nilai error semakin kecil dengan nilai rata-rata error = 10,78% dan hasil trip learning ANFIS sudah mulai mendekati trip target. Pengujian selanjutnya akan dilakukan dengan menaikkan nilai MF menjadi 10 mengacu pada pengujian data normal inverse pada gambar 4.6, dimana hasil learning ANFIS menunjukkan nilai yang sesuai dengan target. Berikut hasil plot learning ANFIS pada interface GUI :





**Gambar 4.12** Hasil plot kurva learning ANFIS MF = 10

Pada gambar 4.12 menunjukkan dengan penambahan nilai MF hasil learning ANFIS sudah mencapai target, namun hasil learning ANFIS masih mempunyai error dengan nilai yang sangat kecil. Dengan penambahan nilai MF mulai dari 3, 5 sampai 10 nilai error yang dihasilkan semakin kecil, bahkan pada MF = 10 learning ANFIS sudah mencapai target dengan nilai error yang sangat kecil. Berikut hasil error perbandingan nilai learning ANFIS dengan hasil data normal inverse :

**Tabel 4.6** Hasil error perbandingan trip ANFIS dan trip target dengan MF 10

No.	Arus (s)	Trip Anfis (s)	Trip Target (s)	Error (%)
1	95	28,49401432	28,494	5,02451E-05
2	145	7,768764966	7,769	0,003025369
3	195	5,133513174	5,134	0,009483298
4	245	4,065627127	4,066	0,009171362
5	295	3,474796269	3,475	0,005863093
6	345	3,094660173	3,095	0,01098109
7	395	2,826400358	2,826	0,014164957
8	445	2,625541981	2,626	0,017444731
9	495	2,468422113	2,468	0,017100499

**Lanjutan Tabel 4.6** Hasil error perbandingan trip ANFIS dan trip target dengan MF 10

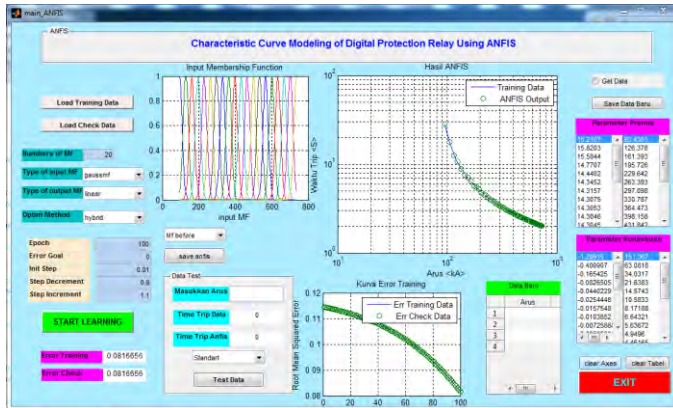
No.	Arus (s)	Trip Anfis (s)	Trip Target (s)	Error (%)
10	545	2,341645635	2,342	0,015133145
11	595	2,236519052	2,237	0,021504329
12	645	2,14779006	2,148	0,009774685
13	695	2,071507035	2,072	0,023797395
14	735	2,017928749	2,018	0,003530907

Dari hasil tabel 4.6 dengan pengujian MF=10 didapatkan nilai error yang sangat kecil dengan nilai rata-rata dibawah 0,1% lebih tepatnya 0,01% , dimana hasil tersebut menunjukkan learning ANFIS sudah mencapai target dan bentuk kurva learning ANFIS sama dengan kurva data normal inverse seperti ditunjukkan pada gambar 4.12. Untuk pengujian kali ini dengan mengganti nilai MF mulai dari terkecil yaitu 3 sampai nilai MF yang mendekati jumlah data (14 data) yaitu MF = 10, hasil error berbanding terbalik dengan nilai membership function (MF) yang artinya jika jumlah MF semakin besar atau mendekati jumlah data target nilai dari error semakin kecil dengan hasil learning ANFIS semakin mendekati target.

#### 4.4 Pengambilan Data Non-Standart

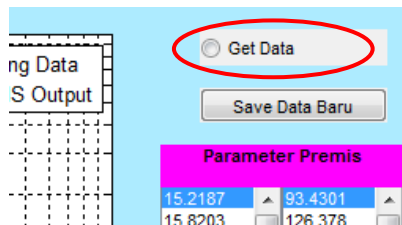
Untuk membuat kurva non-standart sebelumnya harus dilakukan pengambilan data baru pada plot kurva di interface GUI dengan cara mengambil titik-titik data baru sesuai keinginan user. Titik data baru tersebut harus dengan tujuan kurva tidak bersinggungan lagi dengan kurva lainnya. Dengan didapatkan titik-titik data baru kurva secara otomatis akan menjadi data non-standart dikarenakan data baru sudah tidak mengikuti perhitungan dari rumus untuk menentukan waktu trip. Berikut merupakan langkah-langkah pengambilan data baru pada interface GUI :

1. Jalankan interface pada GUI dan mulai training data
2. Akan muncul hasil plot kurva hasil learning data seperti gambar dibawah ini :



**Gambar 4.13** Tampilan GUI dengan plot kurva

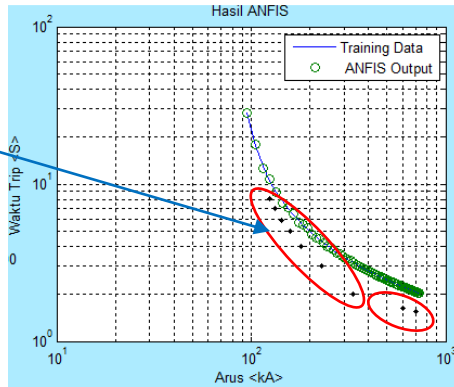
3. Percobaan diatas menggunakan data dengan jarak antar arus 10 A dan jumlah data sebanyak 65 data. Setelah didapatkan plot kurva klik pada tombol di interface yang bertuliskan "get data" seperti ditunjukkan gambar berikut :



**Gambar 4.14** Tombol "get data" pada interface GUI

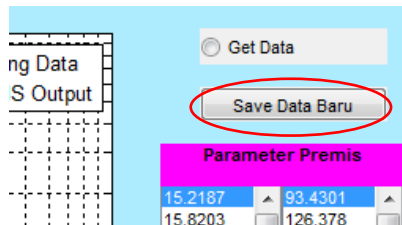
4. Dengan menekan tombol "get data" dapat dilakukan pengambilan titik data baru sesuai dengan yang diinginkan user yang mengacu agar kurva tidak bersinggungan lagi. Pengambilan data baru seperti pada gambar dibawah ini :

**Titik Data Baru**



**Gambar 4.15** Pengambilan titik-titik data baru

5. Data baru tersebut dapat langsung disave dengan format tabel microsoft excel dengan menekan tombol "save data baru" seperti gambar dibawah ini :



**Gambar 4.16** Tombol save data baru di interface GUI

Penyimpanan data baru dengan menggunakan format Ms.Excel bertujuan untuk memudahkan user menggabungkan data yang lama yang sudah dalam format Ms.Excel. Dan gambar selanjutnya menunjukkan hasil data baru dalam bentuk tabel Ms. Excel :

**Tabel 4.7** Tabel data baru dalam bentuk excel

<b>No.</b>	<b>Arus (A)</b>	<b>Trip (s)</b>
<b>1</b>	130,9157	9,019291
<b>2</b>	152,7017	5,968457
<b>3</b>	169,2046	5,011872
<b>4</b>	192,3635	4,012807
<b>5</b>	242,3266	3,01516
<b>6</b>	301,376	2,53191
<b>7</b>	394,5484	2,092617
<b>8</b>	497,0257	1,872465
<b>9</b>	602,4823	1,757226
<b>10</b>	704,3	1,74

## **4.5 Pengujian Data Kurva Non-Standart**

### **4.5.1 Pembuatan data non-standart**

Pada percobaan kali ini akan dilakukan terlebih dahulu pembuatan data Non-standart dengan menggabungkan data normal inverse pada rele yang sudah ada dengan data-data baru yang didapatkan. Penggabungan data dengan melihat batas dimana nilai pengambilan data baru akan menggantikan data lama, semisalkan data baru dimulai pada nilai arus 130,91 A seperti pada tabel 4.9 maka pada data lama nilai dibawah 125 A seperti pada tabel 4.8 akan digantikan dengan data baru. Sehingga penggabungan data tersebut menghasilkan data non-standart dan secara otomatis plot kurva rele menjadi non-standart. Sebagai contoh pembuatan data non-standart dengan menggunakan data kurva standart sebanyak 65 data dan data baru sebanyak 10 titik data. Berikut ini cara penggabungan data kurva standart dan data baru :

**Tabel 4.8** Data Kurva Normal Inverse

Arus (A)	Trip (s)	Arus (A)	Trip (s)	Arus (A)	Trip (s)	Arus (A)	Trip (s)
95	28,494	285	3,571	485	2,497	685	2,086
105	17,488	295	3,475	495	2,468	695	2,072
115	12,937	305	3,386	505	2,441	705	2,058
125	10,442	315	3,305	515	2,415	715	2,044
135	8,862	325	3,230	525	2,389	725	2,031
145	7,769	335	3,160	535	2,365	735	2,018
155	6,966	345	3,095	545	2,342		
165	6,350	355	3,034	555	2,319		
175	5,861	365	2,977	565	2,297		
185	5,464	375	2,924	575	2,276		
195	5,134	385	2,874	585	2,256		
205	4,855	395	2,826	595	2,237		
215	4,616	405	2,782	605	2,218		
225	4,408	415	2,740	615	2,199		
235	4,226	425	2,700	625	2,182		
245	4,066	435	2,662	635	2,164		
255	3,922	445	2,626	645	2,148		
265	3,793	455	2,591	655	2,132		
275	3,677	465	2,558	665	2,116		
		475	2,527	675	2,101		

Selanjutnya dengan mengikuti cara sebelumnya akan diambil data baru pada interface GUI dan data baru tersebut disimpan menggunakan format .xlsx (Ms.excel). Berikut merupakan contoh data baru yang didapatkan sebanyak 10 data :

**Tabel 4.9** Data baru sebanyak 10 data

No.	Arus (A)	Trip (s)
1	130,9157	9,019291
2	152,7017	5,968457
3	169,2046	5,011872
4	192,3635	4,012807
5	242,3266	3,01516

**Lanjutan Tabel 4.9** Data baru sebanyak 10 data

No.	Arus (A)	Trip (s)
6	301,376	2,53191
7	394,5484	2,092617
8	497,0257	1,872465
9	602,4823	1,757226
10	704,3	1,74

Setelah didapatkan data baru seperti tampak pada tabel 4.9, dengan melihat data pertama pada nilai 130,9157A maka pada data normal inverse (standart IEC) penggabungan dimulai pada angka 135 A sehingga pada nilai arus 135 A sampai 735 A akan diganti dengan nilai data baru yang dimulai pada nilai 130,9157A sampai 704,3A. Hasil dari penggabungan data dapat dilihat pada tabel sebagai berikut :

**Tabel 4.10** Data non-standart (penggabungan data standart dan data baru)

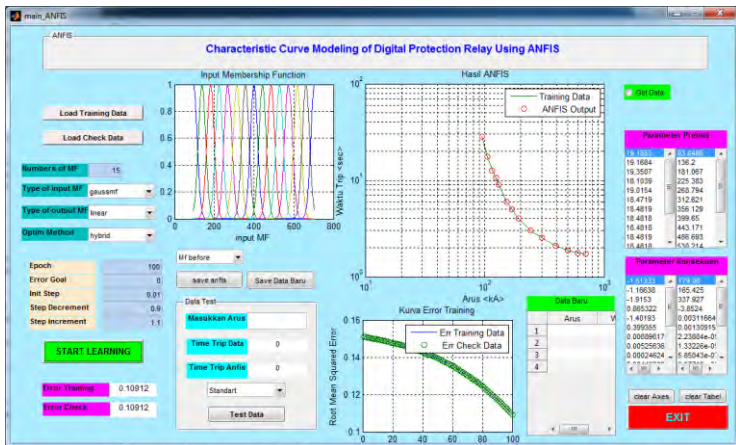
No.	Arus (A)	Trip (s)
1	95	28,494
2	105	17,488
3	115	12,937
4	125	10,442
5	130,9157	9,019291
6	152,7017	5,968457
7	169,2046	5,011872
8	192,3635	4,012807
9	242,3266	3,01516
10	301,376	2,53191
11	394,5484	2,092617
12	497,0257	1,872465
13	602,4823	1,757226
14	704,3	1,74

Dengan data yang baru tersebut seperti pada tabel 4.10 akan dilakukan learning dengan ANFIS untuk membuat kurva non-standart. Pengujian

akan dilakukan pada interface GUI dimana akan menampilkan bentuk kurva non-standart dalam kolom plot.

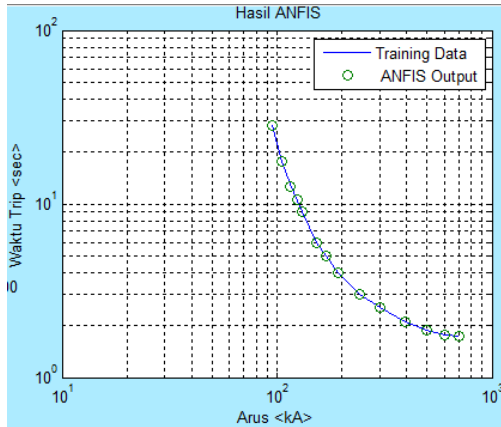
#### 4.5.2 Pengujian Learning ANFIS Data Non-Standart

Dalam pengujian ini akan dilakukan proses learning dengan data non-standart dimana data merupakan penggabungan dari data kurva inverse standart dan data baru. Dengan tujuan apakah data non-standart sudah membentuk kurva yang diinginkan user (kurva tidak bersinggungan). Dan pengujian keakuratan dilakukan dengan menentukan nilai MF (membership function) dan nilai epoch (iterasi) sesuai dengan jumlah data non-standart pada tabel 4.10 dan dilakukan testing beberapa data dengan melihat hasil trip CB (*Circuit Breaker*). Berikut ini tampilan pada interface GUI untuk pengujian data non-standart :



Gambar 4.17 Pengujian data non-standart pada interface GUI





**Gambar 4.18** Tampilan kurva non-standart pada kolom plot interface GUI

Dari melihat pada gambar diatas dengan jumlah 14 data yang ditraining dan jumlah MF (*Membership Function*) yang ditentukan sebesar 10 MF dan jumlah epoch (iterasi) sebanyak 100 kali didapatkan hasil output ANFIS sudah memenuhi target dengan MSE (mean squared error) sebesar 0,739 dan hasil kurva yang didapatkan adalah kurva rele non-standart. Selanjutnya dilakukan pengujian test data dengan mengambil beberapa data sesuai pada tabel 4.10 data non-standart dan beberapa data diluar data non-standart. Pada gambar selanjutnya merupakan pengujian test data dengan tujuan melihat hasil output anfis dibandingkan dengan data non-standart yang sudah ada, berikut gambar pengujian data test :

**Gambar 4.19** Pengujian hasil trip dengan arus 130,9157 A

Pada hasil yang ditunjukkan, waktu trip output ANFIS sebesar 9,22 s sedangkan pada data non-standart seperti pada tabel 4.10 waktu untuk trip pada arus 130,9157 A sebesar 9,01 s. Dari hasil tersebut masih terdapat error dengan nilai sebesar 2,2%. Tetapi untuk rata-rata error keseluruhan mempunyai nilai sebesar 0,739% dimana untuk hasil output ANFIS data lainnya sudah mendekati target. Berikut ini merupakan tabel hasil perbandingan trip output ANFIS dan output data non – standart :

**Tabel 4.11** Hasil error Trip ANFIS berbanding Trip data non-standart

No.	Arus (A)	Trip N-STD (s)	Trip ANFIS (s)	Error (%)
1	95	28,494	27,83509103	2,367548
2	105	17,488	18,79170054	6,940292
3	115	12,937	12,58335434	2,807554
4	125	10,442	9,765716843	6,922094
5	130,9157	9,019291	9,222980113	2,208492
6	152,7017	5,968457	6,393148057	6,64291
7	169,2046	5,011872	4,728081393	6,002243
8	192,3635	4,012807	4,051045991	0,943928

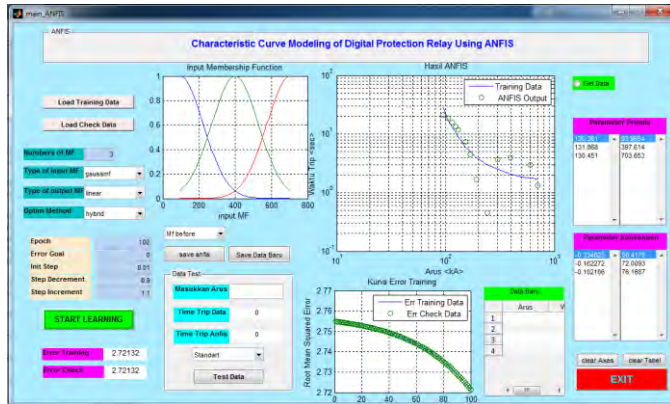
**Tabel Lanjutan 4.11** Hasil error Trip ANFIS berbanding Trip data non-standart

No.	Arus (A)	Trip Target (s)	Trip ANFIS (s)	Error (%)
9	242,3266	3,01516	3,014955748	0,00677
10	301,376	2,53191	2,531820115	0,003553
11	394,5484	2,092617	2,10216083	0,454022
12	497,0257	1,872465	1,879449364	0,371598
13	602,4823	1,757226	1,761483658	0,241717
14	704,3	1,74	1,739353414	0,037174

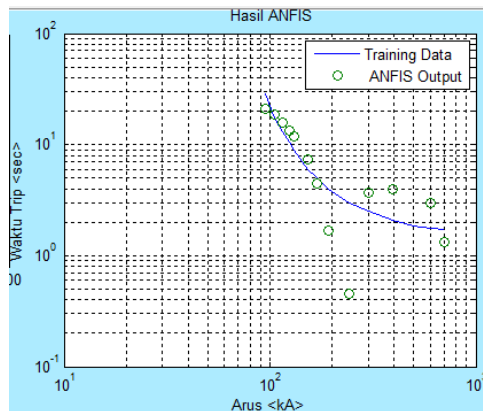
Dari hasil tabel dapat dikatakan pada data-data pertama error masih terlalu besar untuk mencapai target sesuai data non-standart sedangkan untuk data menuju angka terbesar yaitu 192,36A sampai 704,3 A error mulai menunjukkan angka yang kecil dan hasil trip ANFIS bisa dibilang mendekati hasil dari data non- standart (data target).

#### **4.5.3 Pengujian keakuratan dengan mengganti jumlah Membership Function (MF)**

Pengujian kali ini dilakukan untuk membuktikan keakuratan learning ANFIS untuk permodelan kurva data non-standart dengan mengubah-ubah nilai membership function (MF). Data yang akan dilakukan pengujian merupakan data non-standart seperti pada tabel 4.10 dengan jumlah data sebanyak 14 data. Percobaan pertama dilakukan pada nilai membership function (MF) = 3 dan dengan nilai epoch konstan 100, dengan melihat hasil plot kurva pada interface GUI dan test data dengan perbandingan hasil waktu trip learning ANFIS dan hasil waktu trip data non-standart yang ditunjukkan pada tabel 4.10. Berikut hasil pengujian pada interface GUI :



**Gambar 4.20** Hasil percobaan MF 3 pada interface GUI



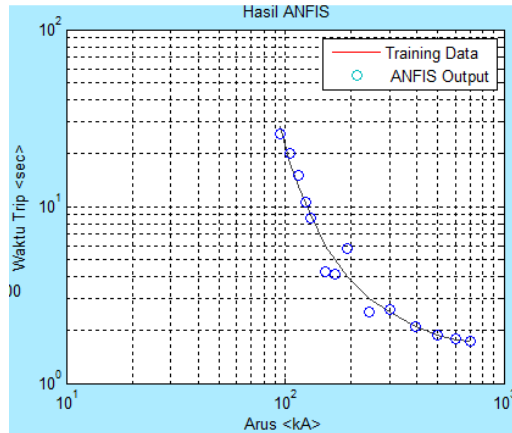
**Gambar 4.21** Hasil plot kurva *learning* ANFIS MF = 3

Dengan pengujian nilai MF = 3 pada gambar 4.21 hasil learning ANFIS yang ditunjukkan lingkaran warna hijau masih jauh untuk memenuhi kurva data target (data non-standart) yang ditampilkan oleh garis warna biru dengan error rata-rata = 139,43%. Berikut error perbandingan hasil learning ANFIS dengan hasil data target (data non-standart) :

**Tabel 4.12** Hasil error perbandingan trip ANFIS dan trip target dengan MF 3

<b>No.</b>	<b>Arus (A)</b>	<b>Trip Target (s)</b>	<b>Trip Anfis (s)</b>	<b>Error (%)</b>
<b>1</b>	95	28,494	21,22720967	34,233847
<b>2</b>	105	17,488	18,46958372	5,3172941
<b>3</b>	115	12,937	15,83027425	18,279124
<b>4</b>	125	10,442	13,32872709	21,660119
<b>5</b>	130,91567	9,019291301	11,92224186	24,349033
<b>6</b>	152,70175	5,968456995	7,296451856	18,200557
<b>7</b>	169,20464	5,011872336	4,474855319	12,000768
<b>8</b>	192,36345	4,012807032	1,682029904	138,5693
<b>9</b>	242,32661	3,015159874	0,449997402	570,0394
<b>10</b>	301,37598	2,531910069	3,704555943	31,654155
<b>11</b>	394,54839	2,092616547	3,930905963	46,765032
<b>12</b>	497,02568	1,872465364	-0,217679965	960,19187
<b>13</b>	602,4823	1,757225861	2,948322886	40,399138
<b>14</b>	704,3	1,74	1,334112855	30,423749

Pengujian selanjutnya menambah nilai MF = 5 dengan nilai epoch tetap 100 konstan. Penambahan nilai MF diharapkan dapat memperbaiki learning untuk mencapai hasil target sesuai dengan data non-standart pada tabel 4.10. Berikut hasil plot pada interface GUI dengan MF = 5 :



**Gambar 4.22** Hasil plot kurva *learning* ANFIS MF = 5

Pada gambar 4.22 menunjukkan dengan penambahan nilai MF hasil learning ANFIS mulai mendekati target dengan error rata-rata = 11,26%, namun hasil learning ANFIS masih belum mencapai data target (data non-standart) dengan adanya error di beberapa titik hasil learning ANFIS. Hasil learning ANFIS dengan penambahan nilai MF = 3 menjadi MF = 5 error yang dihasilkan semakin kecil. Berikut hasil error perbandingan nilai learning ANFIS dengan hasil data target (data non-standart) :

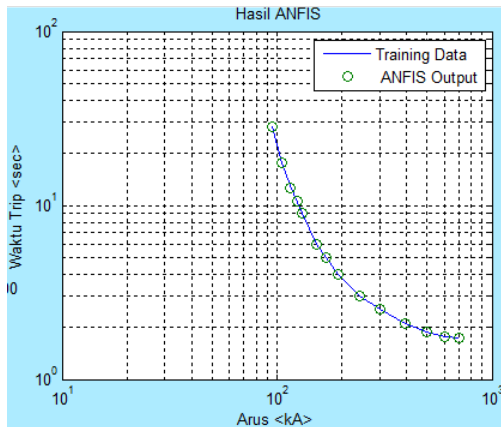
**Tabel 4.13** Hasil error perbandingan trip ANFIS dan trip target dengan MF 5

No.	Arus (A)	Trip Target (s)	Trip Anfis (s)	Error (%)
1	95	28,494	25,5972012	11,317249
2	105	17,488	19,92361337	12,227259
3	115	12,937	14,88201752	13,072009
4	125	10,442	10,64687123	1,9269721
5	130,91567	9,019291301	8,590194555	4,9951924
6	152,70175	5,968456995	4,27947371	39,467079
7	169,20464	5,011872336	4,137708563	21,12676
8	192,36345	4,012807032	5,743322044	30,130907

**Tabel Lanjutan 4.13** Hasil error perbandingan trip ANFIS dan trip target dengan MF 5

No.	Arus (A)	Trip Target (s)	Trip Anfis (s)	Error (%)
9	242,32661	3,015159874	2,526173926	19,356781
10	301,37598	2,531910069	2,597941408	2,5416793
11	394,54839	2,092616547	2,087250013	0,2571103
12	497,02568	1,872465364	1,868250621	0,2255984
13	602,4823	1,757225861	1,776969083	1,1110616
14	704,3	1,74	1,73975976	0,0138088

Pada tabel 4.13 menunjukkan nilai error semakin kecil dan hasil trip learning ANFIS sudah mulai mendekati trip target. Pengujian selajutnya akan dilakukan dengan menaikkan nilai MF menjadi 10 mengacu pada pengujian data non-standart pada gambar 4.18, dimana hasil learning ANFIS menunjukkan nilai yang sesuai dengan target. Berikut hasil plot learning ANFIS pada interface GUI :



**Gambar 4.23** Hasil plot kurva learning ANFIS MF = 10

Pada gambar 4.23 menunjukkan dengan penambahan nilai MF hasil learning ANFIS sudah mencapai target, meskipun hasil learning ANFIS masih mempunyai error dengan nilai yang sangat kecil. Dengan

penambahan nilai MF mulai dari 3, 5 sampai 10 nilai error yang dihasilkan semakin kecil, bahkan pada MF = 10 learning ANFIS sudah mencapai target dengan nilai error yang sangat kecil. Berikut hasil error perbandingan nilai learning ANFIS dengan hasil data target (data non-standart) :

**Tabel 4.14** Hasil error perbandingan trip ANFIS dan trip target dengan MF 10

No.	Arus (A)	Trip Target (s)	Trip Anfis (s)	Error (%)
1	95	28,494	27,83509103	2,3675477
2	105	17,488	18,79170054	6,9402921
3	115	12,937	12,58335434	2,8075544
4	125	10,442	9,765716843	6,9220944
5	130,91567	9,019291301	9,222980113	2,2084924
6	152,70175	5,968456995	6,393148057	6,6429098
7	169,20464	5,011872336	4,728081393	6,0022432
8	192,36345	4,012807032	4,051045991	0,943928
9	242,32661	3,015159874	3,014955748	0,0067704
10	301,37598	2,531910069	2,531820115	0,0035529
11	394,54839	2,092616547	2,10216083	0,4540225
12	497,02568	1,872465364	1,879449364	0,3715982
13	602,4823	1,757225861	1,761483658	0,2417165
14	704,3	1,74	1,739353414	0,0371739

Dari hasil tabel 4.14 dengan pengujian MF=10 didapatkan nilai error yang sangat kecil dengan nilai error rata-rata dibawah 5% lebih tepatnya 2,56%, dimana hasil tersebut menunjukkan learning ANFIS sudah mencapai target dan bentuk kurva learning ANFIS sama dengan kurva data non-standart seperti ditunjukkan pada gambar 4.18. Untuk pengujian kali ini dengan mengganti nilai MF mulai dari terkecil yaitu 3 sampai nilai MF yang mendekati jumlah data (14 data) yaitu MF = 10, hasil error berbanding terbalik dengan nilai membership function (MF) yang artinya jika jumlah MF semakin besar atau mendekati jumlah data



target nilai dari error semakin kecil dengan hasil learning ANFIS semakin mendekati target.

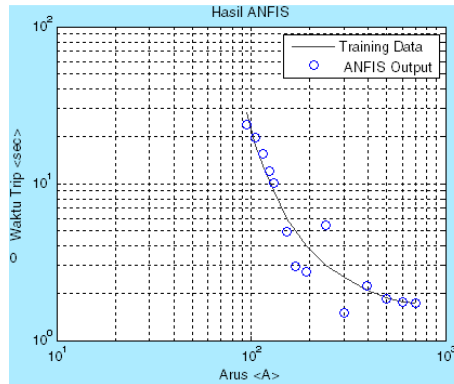
#### 4.5.4 Pengujian keakuratan dengan mengganti jumlah iterasi (Epoch)

Pengujian kali ini dilakukan untuk membuktikan keakuratan learning ANFIS untuk permodelan kurva data non-standart dengan mengubah-ubah nilai iterasi (Epoch). Data yang akan dilakukan pengujian merupakan data non-standart dengan jumlah data sebanyak 14 data seperti pada tabel 4.15 dibawah ini.

**Tabel 4.15** Data percobaan non-standart sebanyak 14 data

No.	Arus (A)	Trip (s)
1	95	28,494
2	105	17,488
3	115	12,937
4	125	10,442
5	130,9157	9,019291
6	152,7017	5,968457
7	169,2046	5,011872
8	192,3635	4,012807
9	242,3266	3,01516
10	301,376	2,53191
11	394,5484	2,092617
12	497,0257	1,872465
13	602,4823	1,757226
14	704,3	1,74

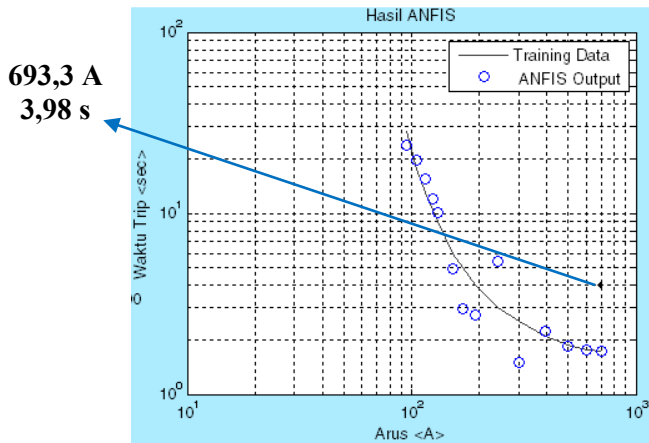
Percobaan pertama dilakukan pada nilai epoch (iterasi) sebanyak 100 dan dengan nilai konstan *membership function* (MF) = 4. Berikut hasil pengujian pada interface GUI :



**Gambar 4.24** Hasil plot kurva learning ANFIS Epoch = 100

Pada pengujian yang dilakukan dengan nilai epoch = 100 didapatkan hasil learning ANFIS yang ditunjukkan lingkaran warna biru pada gambar 4.24 belum bisa mengikuti nilai dari data target yang ditunjukkan garis warna hitam. Pembuktian dapat dilakukan dengan pengujian data test dengan nilai arus diluar data target (data non-standart) diambil contoh nilai arus sebesar 693,3 A (sesuai perpotongan arus kurva R-VCB-9 dan R-ACB-02) didapatkan hasil sebagai berikut :

**Gambar 4.25** Hasil data test arus = 693,3 A, Epoch 100

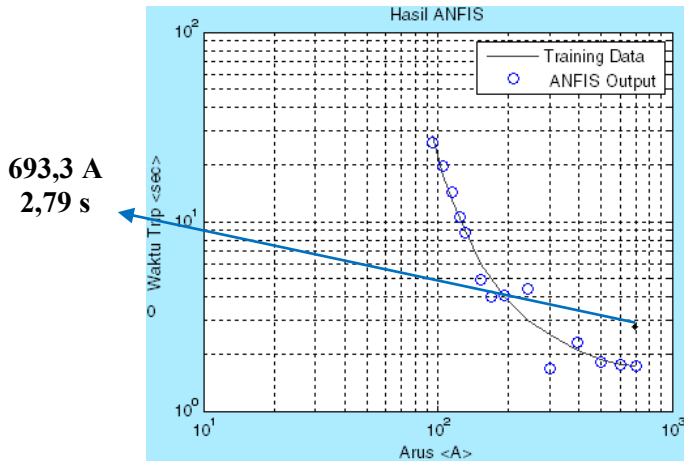


**Gambar 4.26** Pembuktian plot dengan data test arus = 693,3 A, Epoch 100

Dari gambar 4.26 didapatkan hasil learning ANFIS dengan arus 693,3A yang ditunjukkan titik hitam sangat jauh dari target atau tidak sesuai dengan hasil yang diharapkan. Selanjutnya akan dilakukan pengujian dengan nilai epoch (iterasi) = 250 dan berikut hasil yang didapatkan :

The figure shows a "Data Test" window. It has two input fields: "Masukkan Arus" with the value "693.3" and "Time Trip Anfis" with the value "2.79498". Below these is a dropdown menu labeled "Standart" and a button labeled "Test Data".

**Gambar 4.27** Hasil data test arus = 693,3 A, Epoch 250



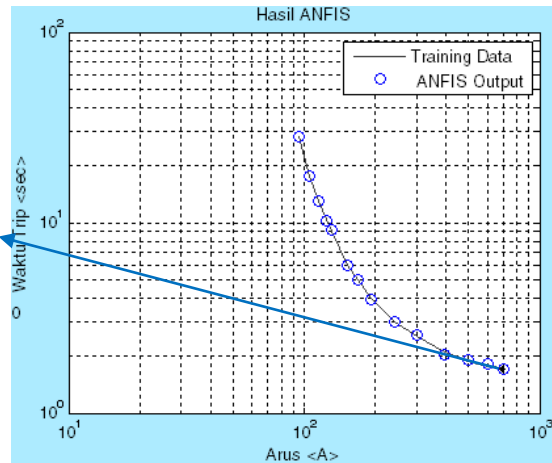
**Gambar 4.28** Pembuktian plot dengan data test arus = 693,3 A, Epoch 250

Dari gambar 4.28 hasil learning ANFIS dengan arus 693,3 A dan epoch 250 masih diluar data target tetapi mengalami perubahan mendekati data target (garais warna hitam) dengan dilakukan pengujian pada kolom data test, hasil waktu trip mengalami perubahan dari 3,98s (100 epoch) menjadi 2,79s (250 epoch). Pengujian berikutnya akan ditambahkan nilai epoch menjadi 500 dengan harapan hasil learning ANFIS sesuai dengan data target.

The figure shows a software interface titled "Data Test". It contains several input fields and a button. The first field is labeled "Masukkan Arus" and has the value "693.3". Below it is a progress bar. The second field is labeled "Time Trip Anfis" and has the value "1.70784". Below that is a dropdown menu with the text "Standart" and a downward arrow. At the bottom is a button labeled "Test Data".

**Gambar 4.29** Hasil data test arus = 693,3 A, Epoch 500

693,3 A  
1,70 s



**Gambar 4.30** Hasil plot dengan data test arus = 693,3A, Epoch 500

Pada gambar 4.30 dapat dilihat learning ANFIS dengan pengujian data test dengan arus 693,3 A sudah sesuai dengan data target (garis warna hitam) yang ditunjukkan titik hitam (menindih garis warna hitam). Sehingga dengan penambahan epoch (iterasi) dapat memperbaiki hasil learning ANFIS sampai mendapatkan error terkecil.

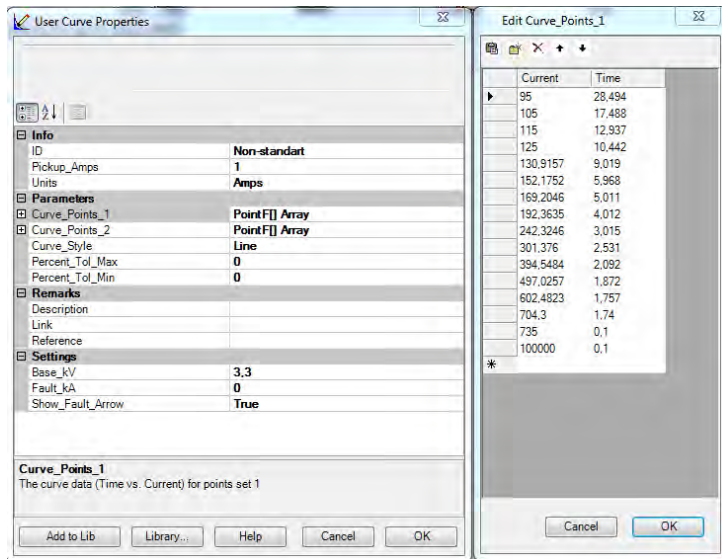
#### 4.6 Pengaplikasian Kurva Pada *Software* Etap 12.6

Sesudah didapatkan data non-standart dan telah dilakukan pengujian learning ANFIS pada interface GUI, akan dilakukan pengaplikasian atau percobaan kurva pada software Etap 12.6 untuk melihat apakah kurva non-standart sudah memenuhi syarat tidak terjadi bersinggungan lagi antar kurva rele dan memudahkan analisa hasil kurva non-standart. Pengaplikasian pada etap 12.6 dapat dilakukan dengan memanfaatkan ruang kerja "*STAR SYSTEM*", dimana dalam ruang kerja tersebut ada menu yaitu "*USER CURVE*" yang dapat membuat suatu kurva sesuai dengan data yang dimasukkan user. Sehingga dengan data non-standart yang didapatkan sebelumnya user dapat membuat kurva baru pada plot Etap 12.6 dan menganalisa kurva baru atau non-standart tersebut. Diketahui data yang akan dijadikan kurva baru atau non-standart pada Etap 12.6 sebagai berikut ini :

**Tabel 4.16** Data non-standart

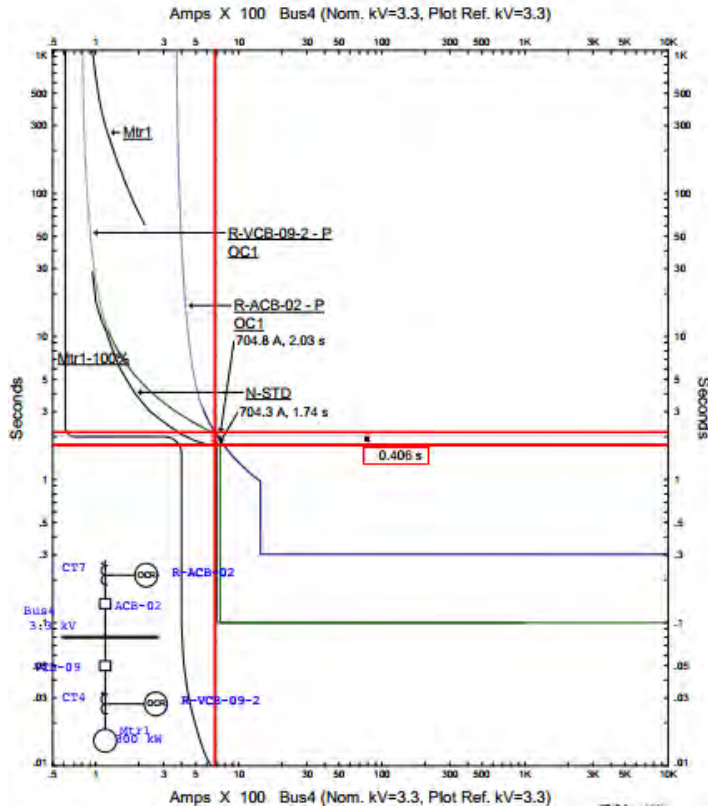
<b>No.</b>	<b>Arus (A)</b>	<b>Trip (s)</b>
<b>1</b>	95	28,494
<b>2</b>	105	17,488
<b>3</b>	115	12,937
<b>4</b>	125	10,442
<b>5</b>	130,9157	9,019291
<b>6</b>	152,7017	5,968457
<b>7</b>	169,2046	5,011872
<b>8</b>	192,3635	4,012807
<b>9</b>	242,3266	3,01516
<b>10</b>	301,376	2,53191
<b>11</b>	394,5484	2,092617
<b>12</b>	497,0257	1,872465
<b>13</b>	602,4823	1,757226
<b>14</b>	704,3	1,74

Data non-standart tersebut akan dimasukkan pada tabel yang ada didalam properties "*USER CURVE*" untuk membentuk kurva baru sesuai dengan data non-standart seperti pada tabel 4.16 diatas. Berikut merupakan gambar tabel dalam properties "*USER CURVE*" :



**Gambar 4.31** Properties "USER CURVE" dan tabel memasukkan data

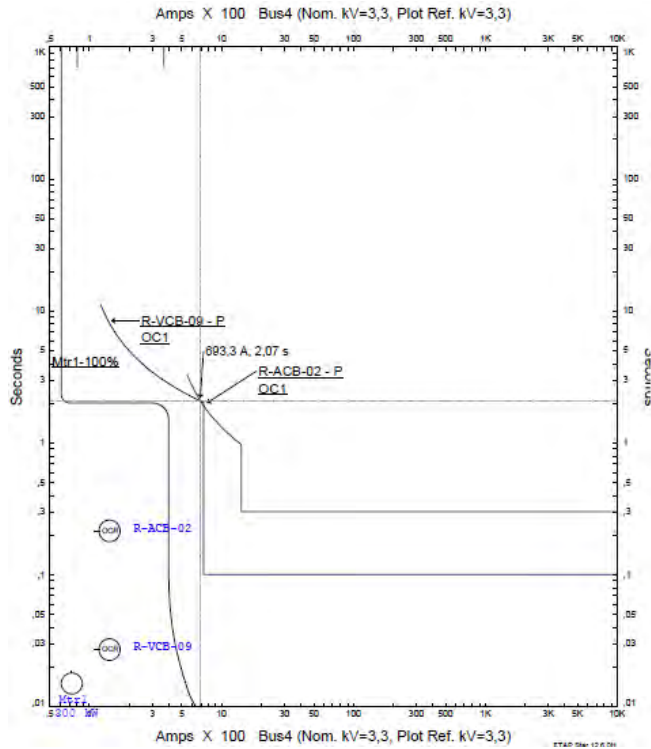
Didalam tabel properties "USER CURVE" sudah terdapat kolom "current" (arus) dan "time" (waktu trip) dimana memudahkan user untuk memasukkan data sesuai dengan data non-standart yang sudah ada. Dengan memasukkan data yang sudah ada Etap 12.6 akan langsung membuat kurva sesuai dengan nilai-nilai data tersebut dimana kurva yang dihasilkan sudah termasuk kurva non-standart. Nilai 735 A dengan waktu trip 0,1 S merupakan data untuk plot bagian definite kurva dan nilai 10000 A dengan 0,1S merupakan batas arus kurva pada bagian definite. Untuk memplot kurva terlebih dahulu jangan lupa untuk mengisi "Base KV" sesuai dengan rate tegangan dimana rele akan diletakkan. Berikut ini merupakan hasil dari plot kurva menggunakan data non-standart :



**Gambar 4.32** Hasil plot data non-standart pada Etap 12.6

Dari hasil plot data non-standart seperti pada gambar 4.32 didapatkan kurva non-standart yang sudah tidak bersinggungan lagi dengan kurva lainnya atau dalam kasus ini dengan kurva rele R-ACB-02. Kasus sebelumnya kurva rele R-VCB-9 bersinggungan dengan kurva rele R-ACB-02 sehingga pada arus sebesar 693,3 A, rele R-VCB-9 dan rele R-ACB-2 mentriapkan CB (*Circuit Breaker*) secara bersamaan pada waktu 2,07s seperti yang ditunjukkan pada gambar berikut ini :



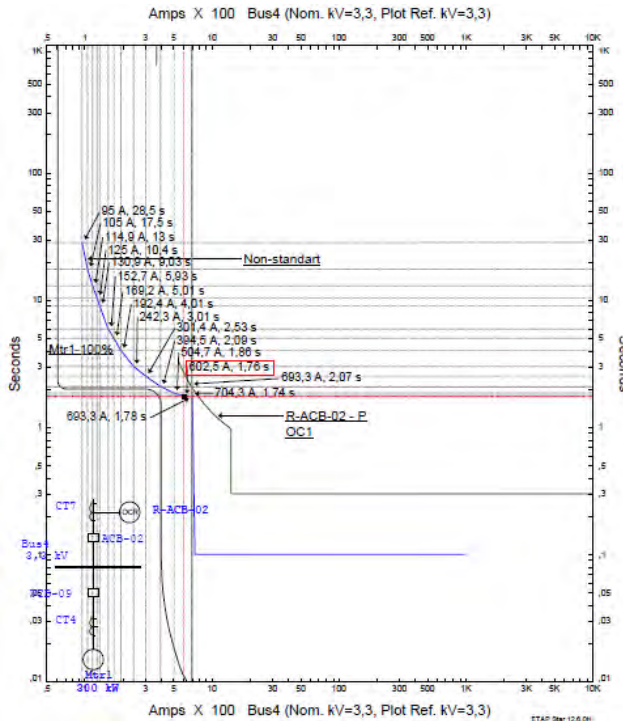


**Gambar 4.33** Plot kurva saat kondisi bersinggungan

Setelah kurva diubah menjadi non-standart seperti pada gambar 4.32 hasil yang didapatkan pada saat arus sebesar 693,3 A, kurva non-standart mentripkan CB selama 1,78s dan pada kurva rele R-ACB-02 mentripkan CB selama 2,07s seperti ditunjukkan gambar 4.32. Dengan begitu rele tidak akan mentripkan CB (*Circuit Breaker*) secara bersamaan pada arus sebesar 693,3 A dan koordinasi dapat berjalan dengan baik. Pembuktian juga dapat dilakukan dengan melihat  $\Delta t$  dari nilai highset kurva non-standart dibandingkan dengan nilai arus yang sama pada kurva R-ACB-02, dengan hasil  $\Delta t = 0,406s$  (standart min  $\Delta t = 0,2$ ) pada nilai arus 704,03 seperti ditunjukkan pada gambar 4.32.

#### 4.6.1 Perbandingan hasil dari data non-standart dan hasil Etap 12.6

Untuk hasil dari software Etap berupa kurva rele non-standart dengan data-data yang dimasukkan seperti pada tabel 4.16 akan dibandingkan ke akuratan waktu trip yang terdapat pada tabel dan hasil kurva dari Etap 12.6. Percobaan akan dilakukan dengan workspace "STAR" di Etap untuk melihat arus dan waktu trip pada kurva non-standart dan dibandingkan hasilnya dengan yang ada pada tabel 4.16. Berikut merupakan gambar pada workspace "STAR" untuk melihat hasil arus dan waktu pada kurva non-standart :



Gambar 4.34 Hasil waktu trip data non-standart pada Etap 12.6

Dengan melihat hasil data non-standart yang di plot pada Etap 12.6, nilai trip yang dihasilkan tidak jauh beda dengan hasil data non-standart pada tabel 4.16. Berikut tabel yang menunjukkan error antara hasil data pada tabel 4.16 dan hasil plot pada etap 12.6 :

**Tabel 4.17** Error perbandingan hasil Etap dan data non-standart

<b>No.</b>	<b>Arus (A)</b>	<b>Trip ANFIS (s)</b>	<b>Trip Etap (s)</b>	<b>Error (%)</b>
<b>1</b>	95	27,835091	28,5	2,3887437
<b>2</b>	105	18,791701	17,5	6,873782
<b>3</b>	115	12,583354	13	3,3110858
<b>4</b>	125	9,7657168	10,4	6,4949984
<b>5</b>	130,91567	9,2229801	9,03	2,0923835
<b>6</b>	152,70175	6,3931481	5,93	7,2444444
<b>7</b>	169,20464	4,7280814	5,01	5,9626428
<b>8</b>	192,36345	4,051046	4,01	1,0132196
<b>9</b>	242,32661	3,0149557	3,01	0,1643722
<b>10</b>	301,37598	2,5318201	2,53	0,0718896
<b>11</b>	394,54839	2,1021608	2,09	0,5784919
<b>12</b>	497,02568	1,8794494	1,86	1,0348437
<b>13</b>	602,4823	1,7614837	1,76	0,0842278
<b>14</b>	704,3	1,7393534	1,74	0,0371739

Dari error yang didapatkan hasil plot kurva pada Etap sudah mendekati target dari data non-standart tabel 4.16 dengan nilai error yang kecil sekali (error rata-rata = 2,66%). Dengan begitu kurva non-standart dapat diaplikasikan untuk simulasi pada software Etap dengan kasus jika koordinasi rele mengalami persinggungan antar rele dan menggunakan beberapa cara untuk mengatasi persinggungan tersebut tidak berhasil, contohnya dengan cara mengganti model kurva dari normal inverse menjadi extreme inverse tidak berhasil dan cara yang lainnya seperti menseting nilai tds tidak juga berhasil disini permodelan kurva rele sangat disarankan untuk menjadi solusinya.

#### 4.7 Pengujian pada *Hardware (Digital Relay Protection)*

Pada bagian ini akan dilakukan pengujian pada hardware, dimana hardware yang akan digunakan sebagai prototype rele adalah mikrokontroller dengan tampilan LCD grafik. Pengujian ini dilakukan menggunakan USB serial untuk input output data. Untuk input pada hardware kali ini berupa data arus IL/ratioCT karena untuk hasil perbandingan IL/ratioCT terlalu besar dengan mempertimbangkan kapasitas dari *injection current* hanya sampai 10A, maka untuk arus masukan digantikan dengan IL/Iset sedangkan untuk output berupa data waktu trip. Pengujian ini bertujuan hanya untuk mengecek hasil kurva yang menunjukkan sebuah kerja rele. Berikut ini langkah-langkah dalam pengujian hardware :

- a) Menyiapkan data yang akan diujikan pada hardware, berikut ini data asli masih dalam satuan ampere (A) :

**Tabel 4.18** Data non-standart untuk pengujian hardware

No.	Arus (A)	Trip (s)
1	95	28.49
2	159.57	5.12
3	221.94	2.78
4	288.36	2.43
5	354.78	2.34
6	420.39	2.12
7	480.33	1.98
8	546.75	1.87
9	609.12	1.81
10	704,3	1.74

Dari data pada tabel akan diubah dalam bentuk (IL/Is) atau nilai arus disisi sekunder dimana Is = 81A, sehingga sebagai contoh pada arus 95A, maka akan didapatkan (IL/Is) :

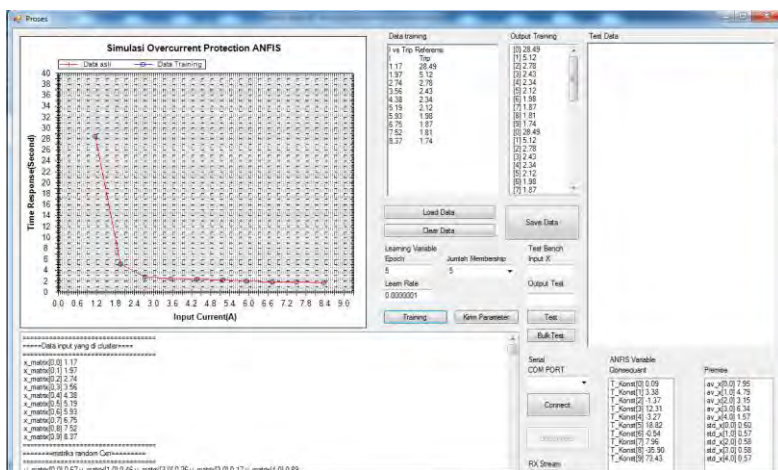
$$IL/I_S = 95/81 = 1,17 A$$

Berikut ini data yang sudah dalam bentuk IL/Is :

**Tabel 4.19** Data non-standart dalam bentuk IL / Is

No.	IL/Is (A)	Trip (s)
1	1.17	28.49
2	1.97	5.12
3	2.74	2.78
4	3.56	2.43
5	4.38	2.34
6	5.19	2.12
7	5.93	1.98
8	6.75	1.87
9	7.52	1.81
10	8.37	1.74

- b) Setelah didapatkan data yang akan diujikan pada hardware, data tersebut akan disimulasikan pada software visualbasic. Seperti pada gambar dibawah ini :



**Gambar 4.35** Interface pada visual basic

Pada gambar diperlihatkan pengujian pada simulasi visual basic yang menghasilkan parameter (parameter premis & konsekuen) hasil learning ANFIS yang dikirimkan ke hardware. Berikut ini merupakan parameter premis dan konsekuen yang didapatkan :

ANFIS Variable	
Qonsequant	Premise
T_Konst[0] 0.09	av_x[0,0] 7.95
T_Konst[1] 3.38	av_x[1,0] 4.79
T_Konst[2] -1.37	av_x[2,0] 3.15
T_Konst[3] 12.31	av_x[3,0] 6.34
T_Konst[4] -3.27	av_x[4,0] 1.57
T_Konst[5] 18.82	std_x[0,0] 0.60
T_Konst[6] -0.54	std_x[1,0] 0.57
T_Konst[7] 7.96	std_x[2,0] 0.58
T_Konst[8] -35.90	std_x[3,0] 0.58
T_Konst[9] 73.43	std_x[4,0] 0.57

**Gambar 4.36** Parameter premis dan konsekuen

- c) Setelah parameter-parameter tersebut dikirimkan ke hardware, dapat dilakukan pengambilan data percobaan dengan cara menginjekkan arus melalui current injector dan melihat waktu trip pada interface LCD 128x64. Berikut ini contoh pengujian pada hardware :



**Current Injector**



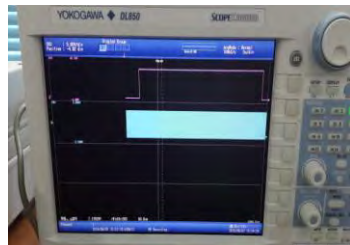
**Multimeter**

**1**



**Hardware (DSP)**

**2**



**Oscilloscope**

**Gambar 4.37** Alur percobaan pada hardware

Percobaan pertama dilakukan dengan memasukkan arus pada current injector dan mencocokkan arus yang dibaca pada hardware, sebagai contoh pada percobaan  $I_L/I_s = 1,17$ , arus yang diberikan current injector sebesar 1,14 A dengan factor error sampai 1,2A. Oleh karena itu arus yang ditunjukkan pada current injector dan pada hardware berbeda. Setelah itu langkah berikutnya dapat dilakukan pengambilan data waktu trip CB dengan menggunakan arus setup pada hardware 1A. Waktu trip dimonitor menggunakan oscilloscope untuk membuktikan bahwa apakah sudah sesuai waktu trip yang ditunjukkan pada interface LCD hardware dan saat CB trip.

Berikut ini hasil perbandingan error antara data percobaan dan hasil pengujian pada hardware :

**Tabel 4.20** Error perbandingan trip ANFIS dan trip Hardware

<b>No.</b>	<b>Arus (A)</b>	<b>Trip Simulasi (s)</b>	<b>Trip Hardware (s)</b>	<b>Error (%)</b>
<b>1</b>	1.17	28.48974	28.376	0.4008317
<b>2</b>	1.97	5.120376	4.92	4.0726829
<b>3</b>	2.74	2.779549	2.78	0.016223
<b>4</b>	3.56	2.430407	2.39	1.6906695
<b>5</b>	4.38	2.339613	2.34	0.0165385
<b>6</b>	5.19	2.120378	2.09	1.4534928
<b>7</b>	5.93	1.979671	1.98	0.0166162
<b>8</b>	6.75	1.870266	1.84	1.6448913
<b>9</b>	7.52	1.809828	1.81	0.0095028
<b>10</b>	8.37	1.740068	1.73	0.5819653

Dari hasil tabel 4.18 pengujian pada hardware dapat diaplikasikan pada hardware (Digital Protection Relay) meskipun pada beberapa data masih terdapat error. Tetapi error tersebut dapat ditoleransi karena masih jauh dibawah nilai 5% lebih tepatnya error rata-rata = 0,7%.



*“halaman ini sengaja dikosongkan”*

## **LAMPIRAN**

```
function varargout = main_ANFIS(varargin)
% MAIN_ANFIS MATLAB code for main_ANFIS.fig
%     MAIN_ANFIS, by itself, creates a new
%     MAIN_ANFIS or raises the existing
%     singleton*.
%
%     H = MAIN_ANFIS returns the handle to a
%     new MAIN_ANFIS or the handle to
%     the existing singleton*.
%
%
%     MAIN_ANFIS('CALLBACK',hObject,eventData,handles,
%     ...) calls the local
%     function named CALLBACK in MAIN_ANFIS.M
%     with the given input arguments.
%
%     MAIN_ANFIS('Property','Value',...)
%     creates a new MAIN_ANFIS or raises the
%     existing singleton*. Starting from the
%     left, property value pairs are
%     applied to the GUI before
%     main_ANFIS_OpeningFcn gets called. An
%     unrecognized property name or invalid
%     value makes property application
%     stop. All inputs are passed to
%     main_ANFIS_OpeningFcn via varargin.
%
%     *See GUI Options on GUIDE's Tools menu.
%     Choose "GUI allows only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to
% help main_ANFIS
```

```

% Last Modified by GUIDE v2.5 23-May-2016
12:13:54

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',           mfilename,
...
                  'gui_Singleton',
gui_Singleton, ...
                  'gui_OpeningFcn',
@main_ANFIS_OpeningFcn, ...
                  'gui_OutputFcn',
@main_ANFIS_OutputFcn, ...
                  'gui_LayoutFcn',    [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback =
str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] =
gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before main_ANFIS is made
visible.
function main_ANFIS_OpeningFcn(hObject,
eventdata, handles, varargin)
% This function has no output args, see
OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a
future version of MATLAB

```

```

% handles      structure with handles and user
data (see GUIDATA)
% varargin     command line arguments to
main_ANFIS (see VARARGIN)

% Choose default command line output for
main_ANFIS
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes main_ANFIS wait for user response
(see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to
the command line.
function varargout =
main_ANFIS_OutputFcn(hObject, eventdata,
handles)
% varargout    cell array for returning output
args (see VARARGOUT);
% hObject      handle to figure
% eventdata    reserved - to be defined in a
future version of MATLAB
% handles      structure with handles and user
data (see GUIDATA)

% Get default command line output from handles
structure
varargout{1} = handles.output;

% --- Executes on button press in
PB_loadDataTrain.

```

```

function PB_loadDataTrain_Callback(hObject,
eventdata, handles)
global myData;
global I;
global trip;
global ttl;

[filename pathname]=uigetfile({'*.xlsx'}, 'File
Selector')
fullpathname=strcat(pathname,filename)
myData=xlsread(fullpathname);
ttl=length(myData);
I=myData(1:ttl,1); %split into
I (1st coloumn) and trip data (2nd coloumn)
trip=myData(1:ttl,2);

% hObject      handle to PB_loadDataTrain (see
GCBO)
% eventdata    reserved - to be defined in a
future version of MATLAB
% handles      structure with handles and user
data (see GUIDATA)

% --- Executes on button press in
PB_loadDataCheck.
function PB_loadDataCheck_Callback(hObject,
eventdata, handles)
global chkData;

[filename pathname]=uigetfile({'*.xlsx'}, 'File
Selector')
fullpathname=strcat(pathname,filename)
chkData=xlsread(fullpathname);

% hObject      handle to PB_loadDataCheck (see
GCBO)

```

```

% eventdata reserved - to be defined in a
future version of MATLAB
% handles structure with handles and user
data (see GUIDATA)

function edit1_Callback(hObject, eventdata,
handles)
% hObject handle to edit1 (see GCBO)
% eventdata reserved - to be defined in a
future version of MATLAB
% handles structure with handles and user
data (see GUIDATA)

% Hints: get(hObject,'String') returns contents
of edit1 as text
% str2double(get(hObject,'String'))
returns contents of edit1 as a double

% --- Executes during object creation, after
setting all properties.
function edit1_CreateFcn(hObject, eventdata,
handles)
% hObject handle to edit1 (see GCBO)
% eventdata reserved - to be defined in a
future version of MATLAB
% handles empty - handles not created until
after all CreateFcns called

% Hint: edit controls usually have a white
background on Windows.
% See ISPC and COMPUTER.
if ispc &&
isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

```

```

% --- Executes on selection change in listBox1.
function listBox1_Callback(hObject, eventdata,
handles)
% hObject      handle to listBox1 (see GCBO)
% eventdata    reserved - to be defined in a
future version of MATLAB
% handles      structure with handles and user
data (see GUIDATA)

% Hints: contents =
cellstr(get(hObject,'String')) returns listBox1
contents as cell array
%           contents{get(hObject,'Value')} returns
selected item from listBox1

% --- Executes during object creation, after
setting all properties.
function listBox1_CreateFcn(hObject, eventdata,
handles)
% hObject      handle to listBox1 (see GCBO)
% eventdata    reserved - to be defined in a
future version of MATLAB
% handles      empty - handles not created until
after all CreateFcns called

% Hint: listbox controls usually have a white
background on Windows.
%           See ISPC and COMPUTER.
if ispc &&
isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in listBox2.

```

```

function listBox2_Callback(hObject, eventdata,
handles)
% hObject      handle to listBox2 (see GCBO)
% eventdata    reserved - to be defined in a
future version of MATLAB
% handles      structure with handles and user
data (see GUIDATA)

% Hints: contents =
cellstr(get(hObject,'String')) returns listBox2
contents as cell array
%           contents{get(hObject,'Value')} returns
selected item from listBox2

% --- Executes during object creation, after
setting all properties.
function listBox2_CreateFcn(hObject, eventdata,
handles)
% hObject      handle to listBox2 (see GCBO)
% eventdata    reserved - to be defined in a
future version of MATLAB
% handles      empty - handles not created until
after all CreateFcns called

% Hint: listbox controls usually have a white
background on Windows.
%           See ISPC and COMPUTER.
if ispc &&
isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in listBox3.
function listBox3_Callback(hObject, eventdata,
handles)

```



```

% hObject      handle to listBox3 (see GCBO)
% eventdata    reserved - to be defined in a
future version of MATLAB
% handles      structure with handles and user
data (see GUIDATA)

% Hints: contents =
cellstr(get(hObject,'String')) returns listBox3
contents as cell array
%           contents{get(hObject,'Value')} returns
selected item from listBox3

% --- Executes during object creation, after
setting all properties.
function listBox3_CreateFcn(hObject, eventdata,
handles)
% hObject      handle to listBox3 (see GCBO)
% eventdata    reserved - to be defined in a
future version of MATLAB
% handles      empty - handles not created until
after all CreateFcns called

% Hint: listBox controls usually have a white
background on Windows.
%           See ISPC and COMPUTER.
if ispc &&
isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata,
handles)
% hObject      handle to edit2 (see GCBO)

```

```

% eventdata reserved - to be defined in a
future version of MATLAB
% handles structure with handles and user
data (see GUIDATA)

% Hints: get(hObject,'String') returns contents
of edit2 as text
% str2double(get(hObject,'String'))
returns contents of edit2 as a double

% --- Executes during object creation, after
setting all properties.
function edit2_CreateFcn(hObject, eventdata,
handles)
% hObject handle to edit2 (see GCBO)
% eventdata reserved - to be defined in a
future version of MATLAB
% handles empty - handles not created until
after all CreateFcns called

% Hint: edit controls usually have a white
background on Windows.
% See ISPC and COMPUTER.
if ispc &&
isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function edit3_Callback(hObject, eventdata,
handles)
% hObject handle to edit3 (see GCBO)
% eventdata reserved - to be defined in a
future version of MATLAB

```

```

% handles      structure with handles and user
data (see GUIDATA)

% Hints: get(hObject,'String') returns contents
of edit3 as text
%          str2double(get(hObject,'String'))
returns contents of edit3 as a double

% --- Executes during object creation, after
setting all properties.
function edit3_CreateFcn(hObject, eventdata,
handles)
% hObject      handle to edit3 (see GCBO)
% eventdata    reserved - to be defined in a
future version of MATLAB
% handles      empty - handles not created until
after all CreateFcns called

% Hint: edit controls usually have a white
background on Windows.
%          See ISPC and COMPUTER.
if ispc &&
isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit4_Callback(hObject, eventdata,
handles)
% hObject      handle to edit4 (see GCBO)
% eventdata    reserved - to be defined in a
future version of MATLAB
% handles      structure with handles and user
data (see GUIDATA)

```

```

% Hints: get(hObject,'String') returns contents
of edit4 as text
%         str2double(get(hObject,'String'))
returns contents of edit4 as a double


% --- Executes during object creation, after
setting all properties.
function edit4_CreateFcn(hObject, eventdata,
handles)
% hObject      handle to edit4 (see GCBO)
% eventdata    reserved - to be defined in a
future version of MATLAB
% handles      empty - handles not created until
after all CreateFcns called


% Hint: edit controls usually have a white
background on Windows.
%         See ISPC and COMPUTER.
if ispc &&
isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function edit5_Callback(hObject, eventdata,
handles)
% hObject      handle to edit5 (see GCBO)
% eventdata    reserved - to be defined in a
future version of MATLAB
% handles      structure with handles and user
data (see GUIDATA)


% Hints: get(hObject,'String') returns contents
of edit5 as text
%         str2double(get(hObject,'String'))
returns contents of edit5 as a double

```

```

% --- Executes during object creation, after
setting all properties.
function edit5_CreateFcn(hObject, eventdata,
handles)
% hObject      handle to edit5 (see GCBO)
% eventdata    reserved - to be defined in a
future version of MATLAB
% handles      empty - handles not created until
after all CreateFcns called

% Hint: edit controls usually have a white
background on Windows.
%           See ISPC and COMPUTER.
if ispc &&
isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit6_Callback(hObject, eventdata,
handles)
% hObject      handle to edit6 (see GCBO)
% eventdata    reserved - to be defined in a
future version of MATLAB
% handles      structure with handles and user
data (see GUIDATA)

% Hints: get(hObject,'String') returns contents
of edit6 as text
%           str2double(get(hObject,'String'))
returns contents of edit6 as a double

% --- Executes during object creation, after
setting all properties.

```

```

function edit6_CreateFcn(hObject, eventdata,
handles)
% hObject      handle to edit6 (see GCBO)
% eventdata    reserved - to be defined in a
future version of MATLAB
% handles      empty - handles not created until
after all CreateFcns called

% Hint: edit controls usually have a white
background on Windows.
%           See ISPC and COMPUTER.
if ispc &&
isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in
PB_startLearning.
function PB_startLearning_Callback(hObject,
eventdata, handles)
% -----
% STEP-2 : GENERATE STRUCTURE OF FIS
% -----
% 1 input - 1 output
% Parameter : Data Training =>myData
%           Number of Membership Function
[input]=> numMFs
%           Type of membership unction [input]
=>inputMF,
%           [output]=>outputMF
%
% Type of membership function
mfType=trimf,trapmf, gbellmf, gaussmf, gauss2mf,
pimf, designmf, psigmf
% grid partition single-output Sugeno-type FIS
by using grid partitioning on the data

```

```

% format
myFIS=genfis1(data,numMFs,inmftype,outmftype)
% -----
global myData;
global I;
global trip;
global chkData;

numMFs=str2num(get(handles.edit1,'string'))

inpMF=get(handles.popupmenu3,'value');
if inpMF==1
    inputMF=char('trimf')
elseif inpMF==2
    inputMF=char('trapmf')
elseif inpMF==3
    inputMF=char('gbellmf')
elseif inpMF==4
    inputMF=char('gaussmf')
elseif inpMF==5
    inputMF=char('gauss2mf')
elseif inpMF==6
    inputMF=char('pimf')
elseif inpMF==7
    inputMF=char('desigmf')
elseif inpMF==8
    inputMF=char('psigmf')
end

outMF=get(handles.popupmenu4,'value');
if outMF==1
    outputMF=char('linear')
elseif outMF==2
    outputMF=char('constant')
end

myFIS=genfis1(myData,numMFs,inputMF,outputMF);

```

```

% -----
---
% STEP-3 : Train FIS
% -----
---
% [fis,error,stepsize,chkFis,chkErr] =
anfis(trnData,initFIS,trnOpt,dispOpt,chkData,opt
Method)
%
% -----
epoch=str2num(get(handles.edit2,'string'))
errgoal=str2num(get(handles.edit3,'string'))
initsteps=str2num(get(handles.edit4,'string'))
stepdec=str2num(get(handles.edit5,'string'))
stepinc=str2num(get(handles.edit6,'string'))

optM=get(handles.popupmenu5,'value');
if optM==1
    optMethod=1
elseif optM==2
    optMethod=0
end

opttr=[epoch,errgoal,initsteps,stepdec,stepinc];
%[epoch];[error goal];[initial step size];[step
size decrease rate];[step size insrease rate]
dispot=[1,1,1,1]; %display otion
1=display, [ANFIS info];[error];[stepsize at
each parameter update];[final result]
%optMethod=1; %an optional
optimization method used in membership function
parameter training: either 1 for the hybrid
method or 0 for the backpropagation method.
%The hybrid
optimization method is a combination of least-
squares and backpropagation gradient descent
method.

```



```

% execute learning
% -----
[outmyFIS,err1,stepsize,chkFis,chkErr]=anfis(myD
ata,myFIS,opttr,dispot,chkData,optMethod);

% display input MF before and after training
% -----
pilihmfplot=get(handles.popupmenu8,'value');
switch pilihmfplot
    case 1
%figure(1)
%subplot (2,1,1)
[x,mf]=plotmf(myFIS,'input',1);
axes(handles.axes1);
plot(x,mf);
xlabel('input MF');
title('Input Membership Function')
grid on
    case 2
%subplot (2,1,2)
[x,mf]=plotmf(outmyFIS,'input',1);
axes(handles.axes1);
plot(x,mf);
xlabel('input MF');
title('Input Membership Function after
Training')
grid on
end

% -----
---
% STEP-4 : Test FIS
% -----
---

% Display data training and ANFIS result

```

```

% -----
---
%handles.axes2;
%figure(2)
axes(handles.axes3);
loglog(I,trip, '-',I,evalfis(I,outmyFIS), 'O');
legend('Training Data', ' ANFIS Output');
title (' Hasil ANFIS');
xlabel('Arus <kA>');
ylabel('Waktu Trip <sec>');
grid on
hold all

% Display Error RMSE, training data nad check
data
% -----
---
%figure (3)
x2=(1:epoch);
axes(handles.axes4);
plot(x2,err1, '-',x2,chkErr, 'O');
title('Kurva Error Training');
xlabel('Nilai Epoch');
ylabel('Root Mean Squared Error');
legend('Err Training Data', ' Err Check Data');
grid on

premis=[outmyFIS.input.mf.params];
konsek=[outmyFIS.output.mf.params];
qtyprem=length(premis)
qtykons=length(konsek)

for iin = 0:1:(qtyprem/2)-1
    premis1(iin+1)=premis(2*iin+1);
end
for iin = 1:1:qtyprem/2
    premis2(iin)=premis(2*iin);
end
set(handles.listbox5, 'string', premis1')

```

```

set(handles.listbox6,'string', premis2')

for iin = 0:1:(qtyprem/2)-1
    konsek1(iin+1)=konsek(2*iin+1);
end
for iin = 1:1:qtyprem/2
    konsek2(iin)=konsek(2*iin);
end

set(handles.listbox7,'string', konsek1')
set(handles.listbox8,'string', konsek2')

%figure (4)
%gensurf(outmyFIS)
cc=err1(epoch)
set(handles.text12,'string', cc)
cc=chkErr(epoch)
set(handles.text13,'string', cc)
save anfis.mat
%figure (5)
%axes(handles.axes5);
%plotfis (outmyFIS)

% hObject      handle to PB_startLearning (see
GCBO)
% eventdata    reserved - to be defined in a
future version of MATLAB
% handles      structure with handles and user
data (see GUIDATA)

% --- Executes on selection change in
popupmenu3.
function popupmenu3_Callback(hObject, eventdata,
handles)
% hObject      handle to popupmenu3 (see GCBO)
% eventdata    reserved - to be defined in a
future version of MATLAB

```

```

% handles      structure with handles and user
data (see GUIDATA)

% Hints: contents =
cellstr(get(hObject,'String')) returns
popupmenu3 contents as cell array
%           contents{get(hObject,'Value')} returns
selected item from popupmenu3

% --- Executes during object creation, after
setting all properties.
function popupmenu3_CreateFcn(hObject,
 eventdata, handles)
% hObject      handle to popupmenu3 (see GCBO)
% eventdata    reserved - to be defined in a
future version of MATLAB
% handles      empty - handles not created until
after all CreateFcns called

% Hint: popupmenu controls usually have a white
background on Windows.
%           See ISPC and COMPUTER.
if ispc &&
isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in
popupmenu4.
function popupmenu4_Callback(hObject, eventdata,
 handles)
% hObject      handle to popupmenu4 (see GCBO)
% eventdata    reserved - to be defined in a
future version of MATLAB

```

```

% handles      structure with handles and user
data (see GUIDATA)

% Hints: contents =
cellstr(get(hObject,'String')) returns
popupmenu4 contents as cell array
%           contents{get(hObject,'Value')} returns
selected item from popupmenu4

% --- Executes during object creation, after
setting all properties.
function popupmenu4_CreateFcn(hObject,
 eventdata, handles)
% hObject      handle to popupmenu4 (see GCBO)
% eventdata    reserved - to be defined in a
future version of MATLAB
% handles      empty - handles not created until
after all CreateFcns called

% Hint: popupmenu controls usually have a white
background on Windows.
%           See ISPC and COMPUTER.
if ispc &&
isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in
popupmenu5.
function popupmenu5_Callback(hObject, eventdata,
 handles)
% hObject      handle to popupmenu5 (see GCBO)
% eventdata    reserved - to be defined in a
future version of MATLAB

```

```

% handles      structure with handles and user
data (see GUIDATA)

% Hints: contents =
cellstr(get(hObject,'String')) returns
popupmenu5 contents as cell array
%           contents{get(hObject,'Value')} returns
selected item from popupmenu5

% --- Executes during object creation, after
setting all properties.
function popupmenu5_CreateFcn(hObject,
eventdata, handles)
% hObject      handle to popupmenu5 (see GCBO)
% eventdata    reserved - to be defined in a
future version of MATLAB
% handles      empty - handles not created until
after all CreateFcns called

% Hint: popupmenu controls usually have a white
background on Windows.
%           See ISPC and COMPUTER.
if ispc &&
isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject,
eventdata, handles)
% hObject      handle to pushbutton4 (see GCBO)
% eventdata    reserved - to be defined in a
future version of MATLAB
% handles      structure with handles and user
data (see GUIDATA)

```

```
delete(handles.figure1)
```

```
% --- Executes on selection change in listBox5.  
function listBox5_Callback(hObject, eventdata,  
handles)  
% hObject      handle to listBox5 (see GCBO)  
% eventdata    reserved - to be defined in a  
future version of MATLAB  
% handles      structure with handles and user  
data (see GUIDATA)
```

```
% Hints: contents =  
cellstr(get(hObject,'String')) returns listBox5  
contents as cell array  
%           contents{get(hObject,'Value')} returns  
selected item from listBox5
```

```
% --- Executes during object creation, after  
setting all properties.  
function listBox5_CreateFcn(hObject, eventdata,  
handles)
```

```
% hObject      handle to listBox5 (see GCBO)  
% eventdata    reserved - to be defined in a  
future version of MATLAB  
% handles      empty - handles not created until  
after all CreateFcns called
```

```
% Hint: listbox controls usually have a white  
background on Windows.
```

```
%           See ISPC and COMPUTER.
```

```
if ispc &&  
isequal(get(hObject,'BackgroundColor'),  
get(0,'defaultUicontrolBackgroundColor'))  
    set(hObject,'BackgroundColor','white');  
end
```

```

% --- Executes on selection change in listbox6.
function listbox6_Callback(hObject, eventdata, handles)
% hObject      handle to listbox6 (see GCBO)
% eventdata    reserved - to be defined in a
future version of MATLAB
% handles      structure with handles and user
data (see GUIDATA)

% Hints: contents =
cellstr(get(hObject,'String')) returns listbox6
contents as cell array
%           contents{get(hObject,'Value')} returns
selected item from listbox6

% --- Executes during object creation, after
setting all properties.
function listbox6_CreateFcn(hObject, eventdata, handles)
% hObject      handle to listbox6 (see GCBO)
% eventdata    reserved - to be defined in a
future version of MATLAB
% handles      empty - handles not created until
after all CreateFcns called

% Hint: listbox controls usually have a white
background on Windows.
%           See ISPC and COMPUTER.
if ispc &&
isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in radiobutton1.

```



```

function radiobutton1_Callback(hObject,
 eventdata, handles)
% hObject    handle to radiobutton1 (see GCBO)
% eventdata  reserved - to be defined in a
future version of MATLAB
% handles    structure with handles and user
data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle
state of radiobutton1

% --- Executes on button press in radiobutton2.
function radiobutton2_Callback(hObject,
 eventdata, handles)
% hObject    handle to radiobutton2 (see GCBO)
% eventdata  reserved - to be defined in a
future version of MATLAB
% handles    structure with handles and user
data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle
state of radiobutton2


function edit7_Callback(hObject, eventdata,
 handles)
% hObject    handle to edit7 (see GCBO)
% eventdata  reserved - to be defined in a
future version of MATLAB
% handles    structure with handles and user
data (see GUIDATA)

% Hints: get(hObject,'String') returns contents
of edit7 as text
%         str2double(get(hObject,'String'))
returns contents of edit7 as a double

```

```

% --- Executes during object creation, after
setting all properties.
function edit7_CreateFcn(hObject, eventdata,
handles)
% hObject      handle to edit7 (see GCBO)
% eventdata    reserved - to be defined in a
future version of MATLAB
% handles      empty - handles not created until
after all CreateFcns called

% Hint: edit controls usually have a white
background on Windows.
%           See ISPC and COMPUTER.
if ispc &&
isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject,
eventdata, handles)
% hObject      handle to pushbutton5 (see GCBO)
% eventdata    reserved - to be defined in a
future version of MATLAB
% handles      structure with handles and user
data (see GUIDATA)
global myData;
global I;
global trip;
global chkData;

numMFs=str2num(get(handles.edit1,'string'))

inpMF=get(handles.popupmenu3,'value');

```

```

if inpMF==1
    inputMF=char('trimf')
elseif inpMF==2
    inputMF=char('trapmf')
elseif inpMF==3
    inputMF=char('gbellmf')
elseif inpMF==4
    inputMF=char('gaussmf')
elseif inpMF==5
    inputMF=char('gauss2mf')
elseif inpMF==6
    inputMF=char('pimf')
elseif inpMF==7
    inputMF=char('desigmf')
elseif inpMF==8
    inputMF=char('psigmf')
end

outMF=get(handles.popupmenu4,'value');
if outMF==1
    outputMF=char('linear')
elseif outMF==2
    outputMF=char('constant')
end

myFIS=genfis1(myData,numMFs,inputMF,outputMF);

% -----
% STEP-3 : Train FIS
% -----
% [fis,error,stepsize,chkFis,chkErr] =
% anfis(trnData,initFIS,trnOpt,dispOpt,chkData,opt
% Method)
%
% -----
epoch=str2num(get(handles.edit2,'string'))
errgoal=str2num(get(handles.edit3,'string'))

```

```

initsteps=str2num(get(handles.edit4,'string'))
stepdec=str2num(get(handles.edit5,'string'))
stepinc=str2num(get(handles.edit6,'string'))

optM=get(handles.popupmenu5,'value');
if optM==1
    optMethod=1
elseif optM==2
    optMethod=0
end

opttr=[epoch,errgoal,initsteps,stepdec,stepinc];
%[epoch];[error goal];[initial step size];[step
size decrease rate];[step size insrease rate]
dispot=[1,1,1,1]; %display otion
l=display, [ANFIS info];[error];[stepsize at
each parameter update];[final result]
%optMethod=1; %an optional
optimization method used in membership function
parameter training: either 1 for the hybrid
method or 0 for the backpropagation method.
%The hybrid
optimization method is a combination of least-
squares and backpropagation gradient descent
method.

% execute learning
% -----
[outmyFIS,err1,stepsize,chkFis,chkErr]=anfis(myD
ata,myFIS,opttr,dispot,chkData,optMethod);
pilihtest1=get(handles.popupmenu9,'value');
switch pilihtest1
    case 1
        arustest=str2num(get(handles.edit7,'string'));
        ref=(0.14*0.65)/((arustest/81)^0.02-1);
        hasilanfis=evalfis(arustest,outmyFIS);
        set(handles.text20,'string',ref);
        set(handles.text22,'string',hasilanfis);
    case 2

```

```

arustest=str2num(get(handles.edit7,'string'));
ref=-1.05*(log10(arustest))+1.03;
hasilanfis=evalfis(arustest,outmyFIS);
set(handles.text20,'string',ref);
set(handles.text22,'string',hasilanfis);
end
% --- Executes on selection change in
popupmenu8.
function popupmenu8_Callback(hObject, eventdata,
handles)
% hObject      handle to popupmenu8 (see GCBO)
% eventdata    reserved - to be defined in a
future version of MATLAB
% handles      structure with handles and user
data (see GUIDATA)

% Hints: contents =
cellstr(get(hObject,'String')) returns
popupmenu8 contents as cell array
%           contents{get(hObject,'Value')} returns
selected item from popupmenu8

% --- Executes during object creation, after
setting all properties.
function popupmenu8_CreateFcn(hObject,
eventdata, handles)
% hObject      handle to popupmenu8 (see GCBO)
% eventdata    reserved - to be defined in a
future version of MATLAB
% handles      empty - handles not created until
after all CreateFcns called

% Hint: popupmenu controls usually have a white
background on Windows.
%           See ISPC and COMPUTER.
if ispc &&
isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))

```

```

        set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in
popupmenu9.
function popupmenu9_Callback(hObject, eventdata,
handles)
% hObject    handle to popupmenu9 (see GCBO)
% eventdata  reserved - to be defined in a
future version of MATLAB
% handles     structure with handles and user
data (see GUIDATA)

% Hints: contents =
cellstr(get(hObject,'String')) returns
popupmenu9 contents as cell array
%           contents{get(hObject,'Value')} returns
selected item from popupmenu9

% --- Executes during object creation, after
setting all properties.
function popupmenu9_CreateFcn(hObject,
eventdata, handles)
% hObject    handle to popupmenu9 (see GCBO)
% eventdata  reserved - to be defined in a
future version of MATLAB
% handles     empty - handles not created until
after all CreateFcns called

% Hint: popupmenu controls usually have a white
background on Windows.
%           See ISPC and COMPUTER.
if ispc &&
isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```
end
```

```
% --- Executes on button press in radiobutton3.
function radiobutton3_Callback(hObject,
 eventdata, handles)
% hObject    handle to radiobutton3 (see GCBO)
% eventdata  reserved - to be defined in a
future version of MATLAB
% handles    structure with handles and user
data (see GUIDATA)
global tabel;
for n=1:10
%n=Inf;
axes(handles.axes3);
    [x,y,but] = ginput(1);
    loglog(x,y,'.-k');
%for b=1:n
tabel(n,1)=x(1,1);
tabel(n,2)=y(1,1);
    set(handles.uitable1,'data',tabel);
end
axes(handles.axes3);
loglog(x,y,'.-k');
% [x,y]= ginput(5);
% for n=1:5
% tabel(n,1)=x(n,1);
% tabel(n,2)=y(n,1);
% loglog (x,y,'.k');
% end
% Hint: get(hObject,'Value') returns toggle
state of radiobutton3
```

```
% --- Executes on button press in pushbutton6.
function pushbutton6_Callback(hObject,
 eventdata, handles)
global myData;
global I;
```

```

global trip;
global chkData;
global tabel
% savefile='Databaru.xlsx';
% xlswrite(savefile,tabel);
savedata=uiputfile({'*.xlsx'}, 'Save as')
%fullpathname=strcat(pathname,filename);
%savefile=fullpathname;
xlswrite(savedata,tabel);
% hObject      handle to pushbutton6 (see GCBO)
% eventdata    reserved - to be defined in a
future version of MATLAB
% handles      structure with handles and user
data (see GUIDATA)

% --- Executes on selection change in listBox7.
function listBox7_Callback(hObject, eventdata,
handles)
% hObject      handle to listBox7 (see GCBO)
% eventdata    reserved - to be defined in a
future version of MATLAB
% handles      structure with handles and user
data (see GUIDATA)

% Hints: contents =
cellstr(get(hObject,'String')) returns listBox7
contents as cell array
%           contents{get(hObject,'Value')} returns
selected item from listBox7

% --- Executes during object creation, after
setting all properties.
function listBox7_CreateFcn(hObject, eventdata,
handles)
% hObject      handle to listBox7 (see GCBO)
% eventdata    reserved - to be defined in a
future version of MATLAB

```



```

% handles      empty - handles not created until
after all CreateFcns called

% Hint: listbox controls usually have a white
background on Windows.
%      See ISPC and COMPUTER.
if ispc &&
isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in listbox8.
function listbox8_Callback(hObject, eventdata,
handles)
% hObject      handle to listbox8 (see GCBO)
% eventdata    reserved - to be defined in a
future version of MATLAB
% handles      structure with handles and user
data (see GUIDATA)

% Hints: contents =
cellstr(get(hObject,'String')) returns listbox8
contents as cell array
%      contents{get(hObject,'Value')} returns
selected item from listbox8

% --- Executes during object creation, after
setting all properties.
function listbox8_CreateFcn(hObject, eventdata,
handles)
% hObject      handle to listbox8 (see GCBO)
% eventdata    reserved - to be defined in a
future version of MATLAB
% handles      empty - handles not created until
after all CreateFcns called

```

```

% Hint: listbox controls usually have a white
background on Windows.
%      See ISPC and COMPUTER.
if ispc &&
isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton7.
function pushbutton7_Callback(hObject,
 eventdata, handles)
axes (handles.axes3);
grid on
cla

% hObject      handle to pushbutton7 (see GCBO)
% eventdata    reserved - to be defined in a
future version of MATLAB
% handles      structure with handles and user
data (see GUIDATA)

% --- Executes on button press in pushbutton8.
function pushbutton8_Callback(hObject,
 eventdata, handles)
global myData
global tabel
myData=0;
tabel=0;
set(handles.uitable1,'data',myData);
set(handles.uitable1,'data',tabel);
% hObject      handle to pushbutton8 (see GCBO)
% eventdata    reserved - to be defined in a
future version of MATLAB

```

```

% handles      structure with handles and user
data (see GUIDATA)

% --- Executes on button press in pushbutton9.
function pushbutton9_Callback(hObject,
eventdata, handles)
global myData;
global I;
global trip;
global chkData;
global ttl;

load anfis.mat
[a,b]=size(I);
% savefile='Databaru.xlsx';
% xlswrite(savefile,tabel);
tabeltrain(2:a+1,1)=I;
tabeltrain(2:a+1,2)=evalfis(I,outmyFIS);
% a={'Arus (kA)' 'Waktu (s)'};
% b=cellstr(a);
% tabeltrain(1,:)=b;
savedata=uiinputfile({'*.xlsx'},'Save as')
%fullpathname=strcat(pathname,filename);
%savefile=fullpathname;
xlswrite(savedata,tabeltrain);

% hObject      handle to pushbutton9 (see GCBO)
% eventdata    reserved - to be defined in a
future version of MATLAB
% handles      structure with handles and user
data (see GUIDATA)

% --- Executes on button press in pushbutton10.
function pushbutton10_Callback(hObject,
eventdata, handles)
% hObject      handle to pushbutton10 (see GCBO)

```

```

% eventdata reserved - to be defined in a
future version of MATLAB
% handles structure with handles and user
data (see GUIDATA)
global s;
s=serial('COM5')%, 'BaudRate', 9600);%, 'DataBits',
8, 'Parity', {none}, 'StopBits', 1);
fopen(s)

% --- Executes on button press in pushbutton11.
function pushbutton11_Callback(hObject,
eventdata, handles)
% hObject handle to pushbutton11 (see GCBO)
% eventdata reserved - to be defined in a
future version of MATLAB
% handles structure with handles and user
data (see GUIDATA)
global s;
fclose(s)
delete(s)

% --- Executes on button press in pushbutton12.
function pushbutton12_Callback(hObject,
eventdata, handles)
% hObject handle to pushbutton12 (see GCBO)
% eventdata reserved - to be defined in a
future version of MATLAB
% handles structure with handles and user
data (see GUIDATA)
global s;
% fprintf(s, 'test');
ccc=get(handles.edit7, 'string')
fprintf(s, '%s', ccc);

```

```

function edit9_Callback(hObject, eventdata,
handles)
% hObject      handle to edit9 (see GCBO)
% eventdata    reserved - to be defined in a
future version of MATLAB
% handles      structure with handles and user
data (see GUIDATA)

% Hints: get(hObject,'String') returns contents
of edit9 as text
%           str2double(get(hObject,'String'))
returns contents of edit9 as a double

% --- Executes during object creation, after
setting all properties.
function edit9_CreateFcn(hObject, eventdata,
handles)
% hObject      handle to edit9 (see GCBO)
% eventdata    reserved - to be defined in a
future version of MATLAB
% handles      empty - handles not created until
after all CreateFcns called

% Hint: edit controls usually have a white
background on Windows.
%           See ISPC and COMPUTER.
if ispc &&
isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

## **BAB V**

### **PENUTUP**

#### **5.1. Kesimpulan**

Berdasarkan hasil simulasi dan analisis, maka disimpulkan sebagai berikut :

1. Permodelan kurva karakteristik inverse maupun kurva karakteristik non-standart dapat dilakukan dengan metode pembelajaran ANFIS.
2. Semakin besar nilai MF maka tingkat keakuratan semakin bagus, pada pengujian ini MF dengan tingkat keakuratan paling bagus adalah MF = 10 dengan jumlah data sebanyak 14 data dan rata-rata error 2,56% (masih dibawah 5%).
3. Pada pengujian keakuratan dengan mengganti nilai iterasi (epoch), didapatkan nilai iterasi sebanyak 500 iterasi sudah dapat mencapai hasil sesuai dengan target dengan sebelumnya dilakukan pengujian pada jumlah iterasi 100 kali dan 250 kali.
4. Hasil dari learning ANFIS berupa kurva non-standart dapat diaplikasikan pada software Etap 12.6 dengan hasil kurva R-VCB 9 tidak berhimpitan dengan kurva R-ACB 02. Waktu trip pada rele R-ACB-02 didapatkan 2,07 s sedangkan waktu trip pada rele R-VCB 9 sebesar 1,78 s pada arus 693,3 A.
5. Pengaplikasian kurva non-standart dengan jumlah 14 data pada software Etap 12.6 didapatkan rata-rata error sebesar 2,66%.
6. Hasil pengujian pada hardware dan pada simulasi menunjukkan hasil yang sangat mendekati dengan rata-rata error = 0,7%

#### **5.2. Saran**

- Saran yang dapat diberikan berdasarkan penelitian ini adalah perlu dilakukan penelitian lanjutan mengenai permodelan kurva proteksi, semisal kurva non-standart dengan berbagai jenis kurva Inverse. Selain itu dapat dilakukan pengembangan untuk koordinasi proteksinya.
- Mencoba menggunakan metode Neural Network untuk permodelan kurva non-standart.

***“halaman ini sengaja dikosongkan”***

## DAFTAR PUSTAKA

- [1] Ir R.Wahyudi. "Diktat Kuliah Pengaman Sistem Tenaga Listrik", Jurusan Teknik Elektro, Institut Teknologi Sepuluh Nopember, Surabaya.
- [2] Soeprijanto, Adi "Kestabilan Sistem Tenaga Listrik, Diktat Kuliah Analisis Sistem Tenaga Listrik 2", Teknik Elektro Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember Surabaya
- [3] Anderson, P.M, "Power System Protection", John Wiley & Sons, Inc., Canada, Ch. 3, 1998
- [4] "Datasheet Overcurrent Protection Relays", CKR Series.
- [5] Warsito, B. "Kapita Selekta Statistika Neural Network". BP Undip Semarang, 2009.
- [6] Sri Kusumadewi, Sri Hartati "Neuro – Fuzzy Integrasi Sistem Fuzzy dan Jaringan Syaraf". GRAHA ILMU, Yogyakarta, 2006.
- [7] Jang, J.-S. R. and Sun, C.-T.: Neuro-fuzzy and soft computing: a computational approach to learning and machine intelligence, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1997.
- [8] J.S.R. Jang "ANFIS: Adaptive Network Based Fuzzy Inference Systems", IEEE Trans, Syst. Man Cybernet. vol 23, No 3, pp. 665-685, 1993.
- [9] Rachman Halim, Desain Kurva Konvensional Menggunakan Anfis Pada Digital Protection Relay, Teknik Elektro, Fakultas Industri , ITS, Surabaya, 2015.
- [10] Anang Tjahjono, Dimas Okky Anggriawan, Ardyono Priyadi, Margo Pujiانتara and Mauridhi Hery Purnomo, Overcurrent Relay Curve Modeling and Its Application in the Real Industrial Power Systems Using Adaptive Neuro Fuzzy Inference System, ITS, Surabaya, 2015.
- [11] Anang Tjahjono, Ardyono Priyadi, Margo Pujiانتara and Mauridhi Hery Purnomo, Overcurrent Relay Curve Modeling Using Adaptive Neuro Fuzzy Inference System, ITS, Surabaya, 2014.



- [12] O. Arreola Soria\*, A. Conde Enríquez, L.A. Trujillo Guajardo, Overcurrent relay with unconventional curves and its application in industrial power systems, Universidad Autónoma de Nuevo León, Mexico, 2014.
- [13] Timo Keil and Johann Jager, Member, IEEE, Advanced Coordination Method for Overcurrent Protection Relays Using Nonstandart Tripping Characteristics, January 2008

## **BIODATA PENULIS**



Penulis bernama lengkap Erhankana Ardiana Putra dan dilahirkan di Tulungagung pada tanggal 23 September 1992. Penulis memulai pendidikan di SDN PacarKeling VIII Surabaya dan melanjutkan pendidikan SLTP Negeri 29 Surabaya dan SMA Negeri 9 Surabaya. Pada tahun 2010, penulis melanjutkan pendidikan jenjang Diploma III di Politeknik Elektronika Negeri Surabaya dengan konsentrasi pada bidang studi Teknik Elektronika. Pada tahun 2014 penulis berhasil menyelesaikan pendidikan diploma III dan pada saat itu juga melanjutkan pendidikan untuk jenjang sarjana. Pendidikan sarjana ditempuh di Institut Teknologi Sepuluh Nopember Surabaya di jurusan Teknik Elektro dengan konsentrasi pada bidang studi Teknik Sistem Tenaga

Email :

[erhankana14@mhs.ee.its.ac.id](mailto:erhankana14@mhs.ee.its.ac.id)

[erhankana@gmail.com](mailto:erhankana@gmail.com)